Parson's Problem Appliance for Learning Management Systems

Agile Scrum Development Work Log

**Anthony Narlock,
Jaden Rodriguez,
Shen Lua,
Stephanie Ye**

December 2nd, 2022 - December 13th, 2022

Contents

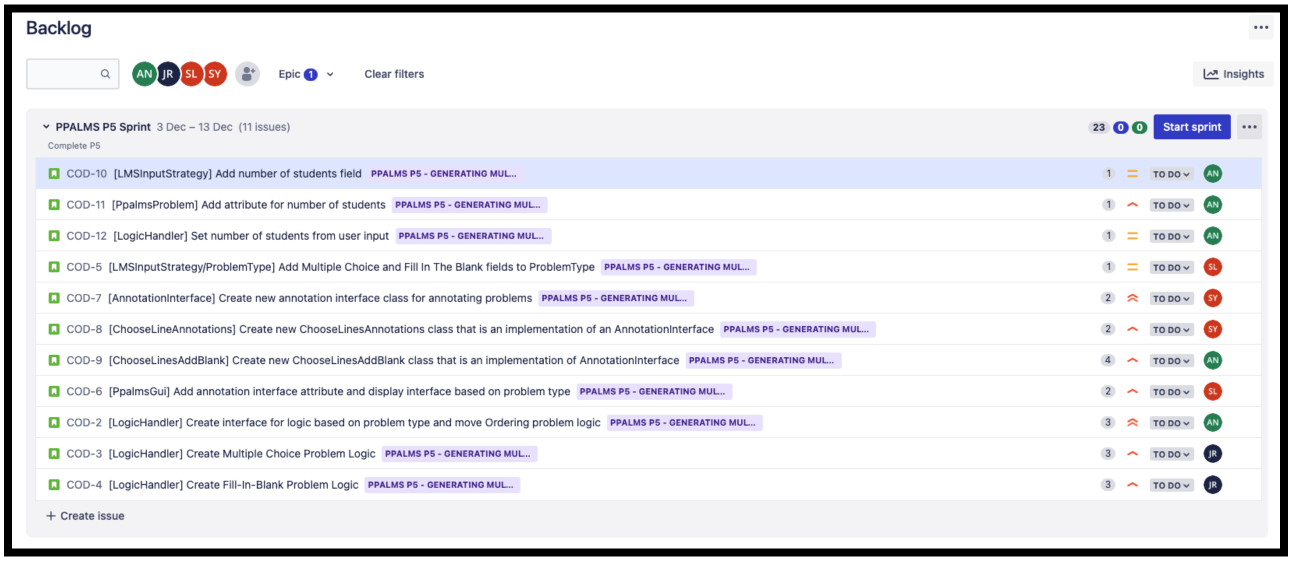
1. Sprint Planning Meeting

Purpose	To choose product-backlog items to accomplish in this iteration and populate the Sprint backlog with subtasks to complete them; to break the tasks into subtasks that can be completed in less than a day; to estimate the amount of work and define "doneness" for each item.
Date of Meeting, Time Duration	December 2nd, 2022 4:00 PM - 5:30 PM (1 hour, 30 minutes)
Meeting Attendance	Anthony Narlock, Jaden Rodriguez, Shen Lua, Stephanie Ye

In this meeting, we planned our time frames. We agreed upon a Sprint from December 3rd to December 13th, and we planned meetup times for the daily SCRUM meetings for the following week. We came up with a story scoring system: 1 means that the task should be short to complete, on the order of 15 minutes; 2 means it should take on the order of 2-3 hours; 3 on the order of 5-6 hours; and 4 on the order of 10 hours of work. This made it so we could plan and distribute tasks accordingly.

In this meeting, we discussed which problems to tackle, namely adding multiple choice and fill-in-the-blank problems to PPALMS. We discussed some design changes using UML, such as a StrategyInterface for the PPALMSGui, and a new AnnotationInterface (JComponent) for the ProblemInputStrategy. This way, the input handler and concrete implementation of the annotation interface to use can be selected based on the problem type chosen by the user. We also discussed implementation details to resolve each problem. For example, for fill-in-the-blank we discussed making 2 combo boxes for selecting indices i and j to indicated what will be “blank”

Before this meeting, Anthony prepared the product backlog on Atlassian’s JIRA agile development tool and added each member. Stories COD-1 through COD-12 were added to the backlog. We made a description for each story following the given story template. *Each of these were added as tasks to our Sprint backlog* to be completed during our sprint and given a score.


Backlog

AN JR SL SY   Epic 1   Clear filters   Insights

PPALMS P5 Sprint 3 Dec – 13 Dec (11 issues)   23 0 0   Start sprint

Complete P5

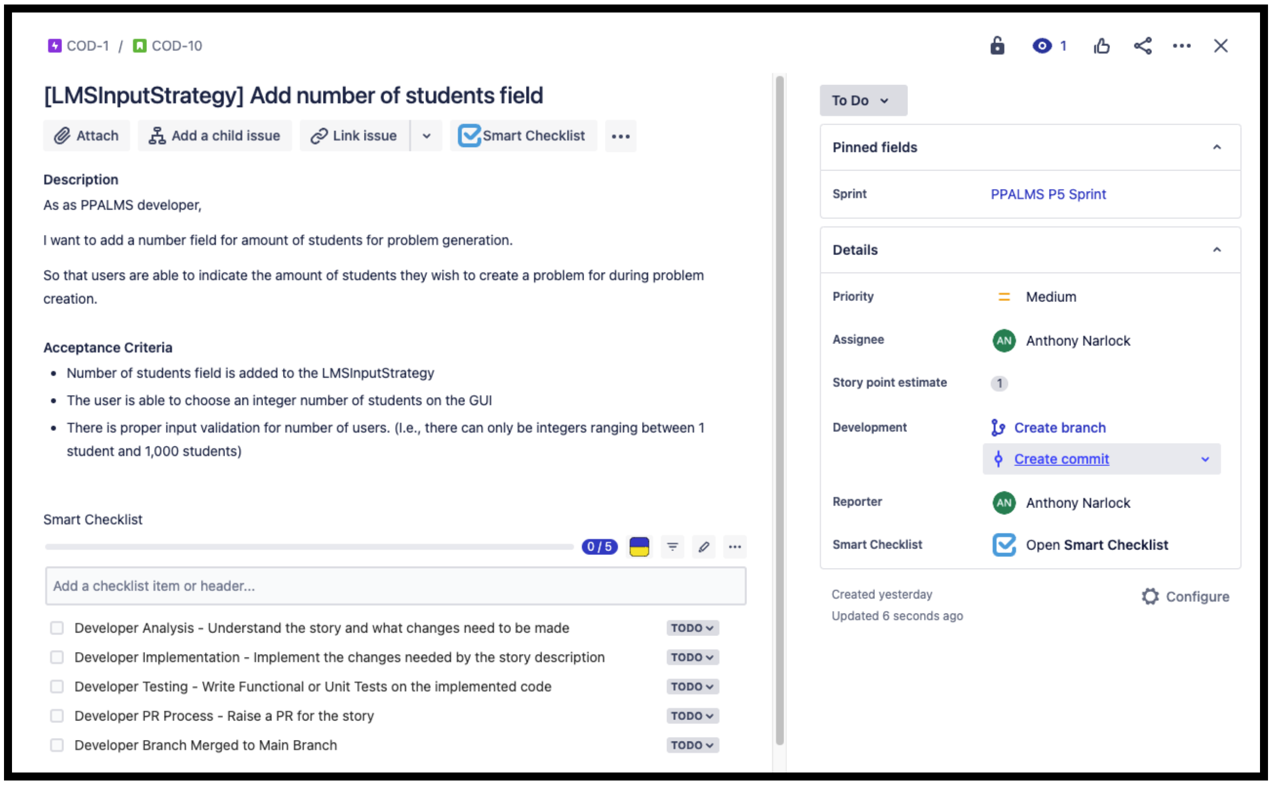
COD-10 [LMSInputStrategy] Add number of students field	PPALMS P5 - GENERATING MUL...	1	=	TO DO	AN
COD-11 [PpalmsProblem] Add attribute for number of students	PPALMS P5 - GENERATING MUL...	1	^	TO DO	AN
COD-12 [LogicHandler] Set number of students from user input	PPALMS P5 - GENERATING MUL...	1	=	TO DO	AN
COD-5 [LMSInputStrategy/ProblemType] Add Multiple Choice and Fill In The Blank fields to ProblemType	PPALMS P5 - GENERATING MUL...	1	=	TO DO	SL
COD-7 [AnnotationInterface] Create new annotation interface class for annotating problems	PPALMS P5 - GENERATING MUL...	2	^^	TO DO	SY
COD-8 [ChooseLineAnnotations] Create new ChooseLinesAnnotations class that is an implementation of an AnnotationInterface	PPALMS P5 - GENERATING MUL...	2	^	TO DO	SY
COD-9 [ChooseLinesAddBlank] Create new ChooseLinesAddBlank class that is an implementation of AnnotationInterface	PPALMS P5 - GENERATING MUL...	4	^	TO DO	AN
COD-6 [PpalmsGui] Add annotation interface attribute and display interface based on problem type	PPALMS P5 - GENERATING MUL...	2	^	TO DO	SL
COD-2 [LogicHandler] Create interface for logic based on problem type and move Ordering problem logic	PPALMS P5 - GENERATING MUL...	3	^^	TO DO	AN
COD-3 [LogicHandler] Create Multiple Choice Problem Logic	PPALMS P5 - GENERATING MUL...	3	^	TO DO	JR
COD-4 [LogicHandler] Create Fill-In-Blank Problem Logic	PPALMS P5 - GENERATING MUL...	3	^	TO DO	JR

+ Create issue

In this meeting, we made a general definition of "done". We define a story as "done" when we meet the following criteria: we have written passing unit tests for each branch or functionally tested each branch of the implementation of the story; the implementation has been reviewed and approved by another developer; the documentation, like JavaDoc comments in the source and unit tests, has been updated; and the execution reports for the tests have been written. This definition was used to make a smart checklist that is used to provide a definition of done for each story as agreed on in our planning meeting in the JIRA software.

We assigned each member tasks in the sprint backlog as a group based on their score, member availability, and their past work. COD-1 is the Epic describing our overarching goal, and was broken down into subtasks.

Anthony was assigned COD-2, and COD-9 through COD-12. Screenshots of their descriptions, acceptance criteria, and other important details are shown below.


COD-1 / COD-10

[LMSInputStrategy] Add number of students field

Attach | Add a child issue | Link issue | Smart Checklist

Description

As as PPALMS developer,

I want to add a number field for amount of students for problem generation.

So that users are able to indicate the amount of students they wish to create a problem for during problem creation.

Acceptance Criteria

- Number of students field is added to the LMSInputStrategy
- The user is able to choose an integer number of students on the GUI
- There is proper input validation for number of users. (I.e., there can only be integers ranging between 1 student and 1,000 students)

Smart Checklist 0 / 5

Add a checklist item or header...

- [ ] Developer Analysis - Understand the story and what changes need to be made TODO
- [ ] Developer Implementation - Implement the changes needed by the story description TODO
- [ ] Developer Testing - Write Functional or Unit Tests on the implemented code TODO
- [ ] Developer PR Process - Raise a PR for the story TODO
- [ ] Developer Branch Merged to Main Branch TODO

To Do

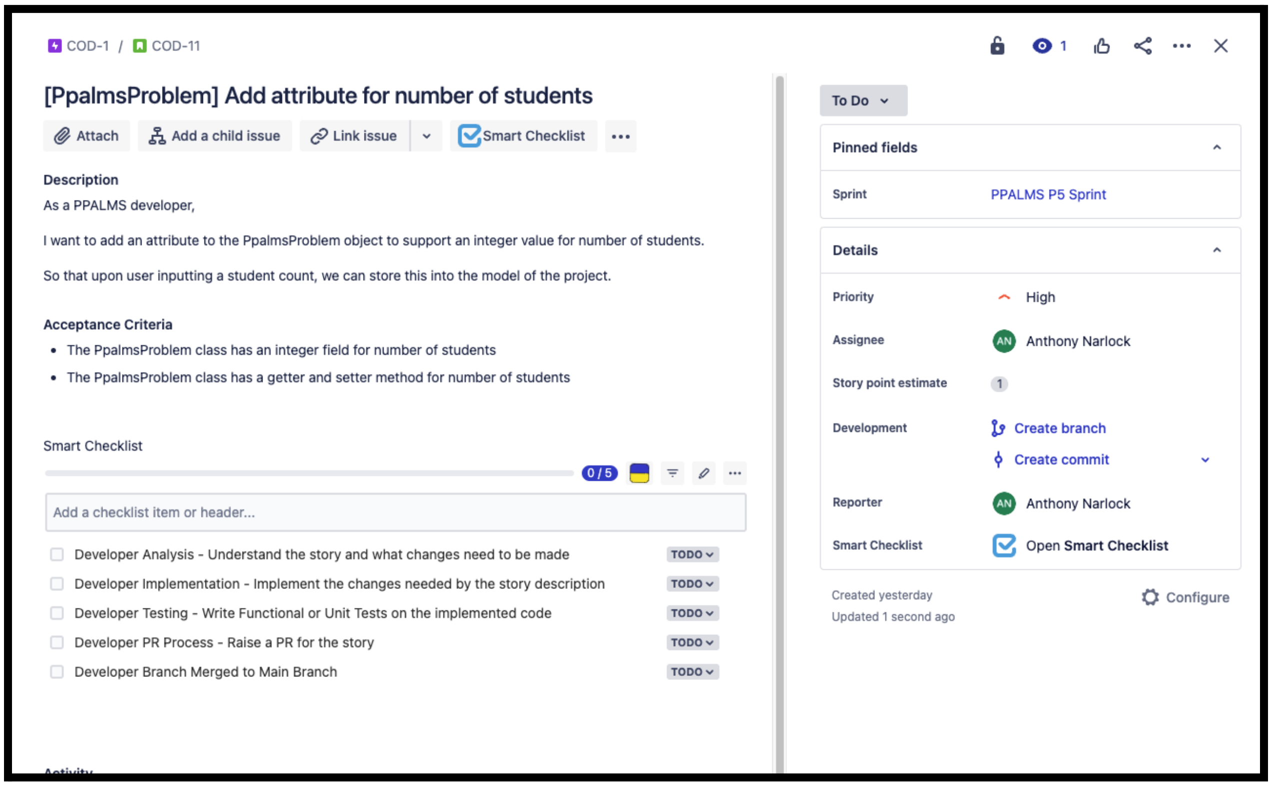
Pinned fields

Sprint	PPALMS P5 Sprint

Details

Priority	Medium
Assignee	Anthony Narlock
Story point estimate	1
Development	Create branch Create commit
Reporter	Anthony Narlock
Smart Checklist	Open Smart Checklist

Created yesterday
Updated 6 seconds ago
Configure


COD-1 / COD-11

[PpalmsProblem] Add attribute for number of students

Attach | Add a child issue | Link issue | Smart Checklist

Description

As a PPALMS developer,

I want to add an attribute to the PpalmsProblem object to support an integer value for number of students.

So that upon user inputting a student count, we can store this into the model of the project.

Acceptance Criteria

- The PpalmsProblem class has an integer field for number of students
- The PpalmsProblem class has a getter and setter method for number of students

Smart Checklist 0 / 5

Add a checklist item or header...

- [ ] Developer Analysis - Understand the story and what changes need to be made TODO
- [ ] Developer Implementation - Implement the changes needed by the story description TODO
- [ ] Developer Testing - Write Functional or Unit Tests on the implemented code TODO
- [ ] Developer PR Process - Raise a PR for the story TODO
- [ ] Developer Branch Merged to Main Branch TODO

To Do

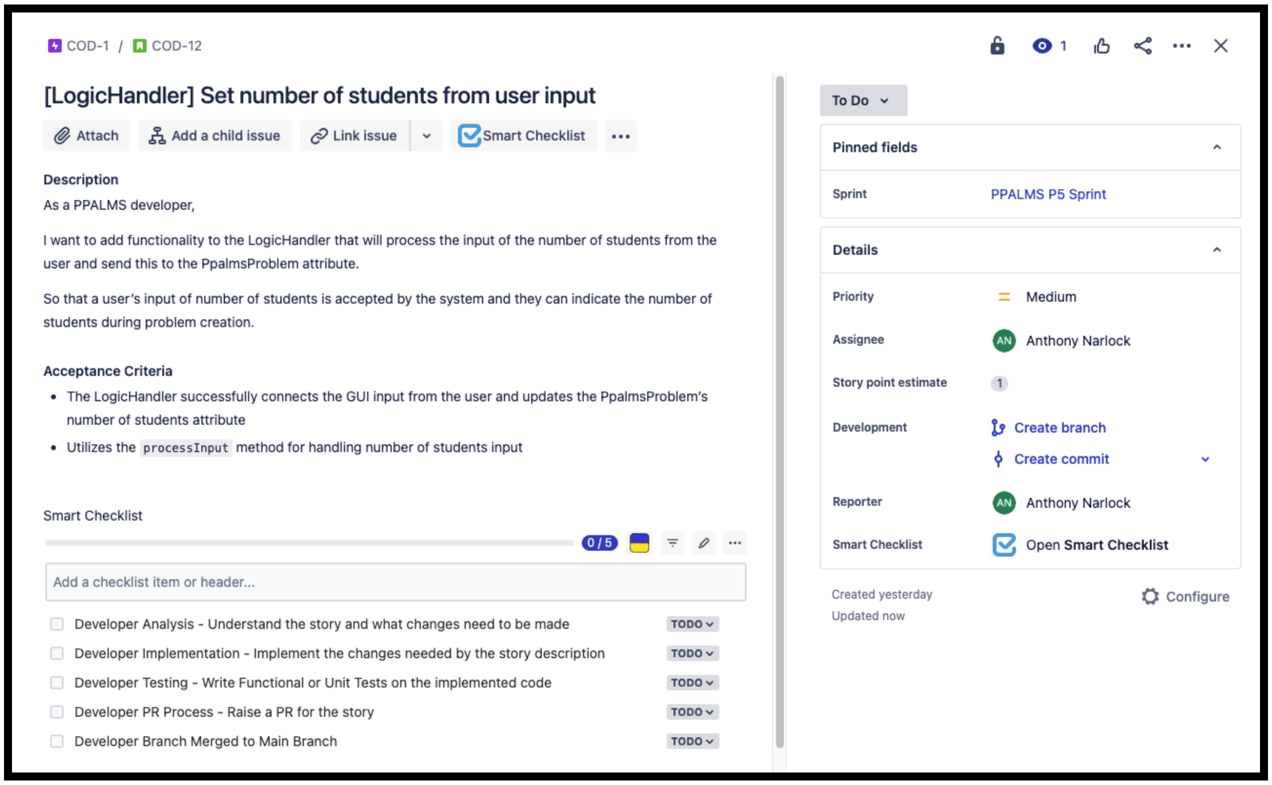
Pinned fields

Sprint	PPALMS P5 Sprint

Details

Priority	High
Assignee	Anthony Narlock
Story point estimate	1
Development	Create branch Create commit
Reporter	Anthony Narlock
Smart Checklist	Open Smart Checklist

Created yesterday
Updated 1 second ago
Configure


COD-1 / COD-12

[LogicHandler] Set number of students from user input

Attach | Add a child issue | Link issue | Smart Checklist

Description

As a PPALMS developer,

I want to add functionality to the LogicHandler that will process the input of the number of students from the user and send this to the PpalmsProblem attribute.

So that a user's input of number of students is accepted by the system and they can indicate the number of students during problem creation.

Acceptance Criteria

- The LogicHandler successfully connects the GUI input from the user and updates the PpalmsProblem's number of students attribute
- Utilizes the `processInput` method for handling number of students input

Smart Checklist 0 / 5

Add a checklist item or header...

- [ ] Developer Analysis - Understand the story and what changes need to be made TODO
- [ ] Developer Implementation - Implement the changes needed by the story description TODO
- [ ] Developer Testing - Write Functional or Unit Tests on the implemented code TODO
- [ ] Developer PR Process - Raise a PR for the story TODO
- [ ] Developer Branch Merged to Main Branch TODO

To Do

Pinned fields

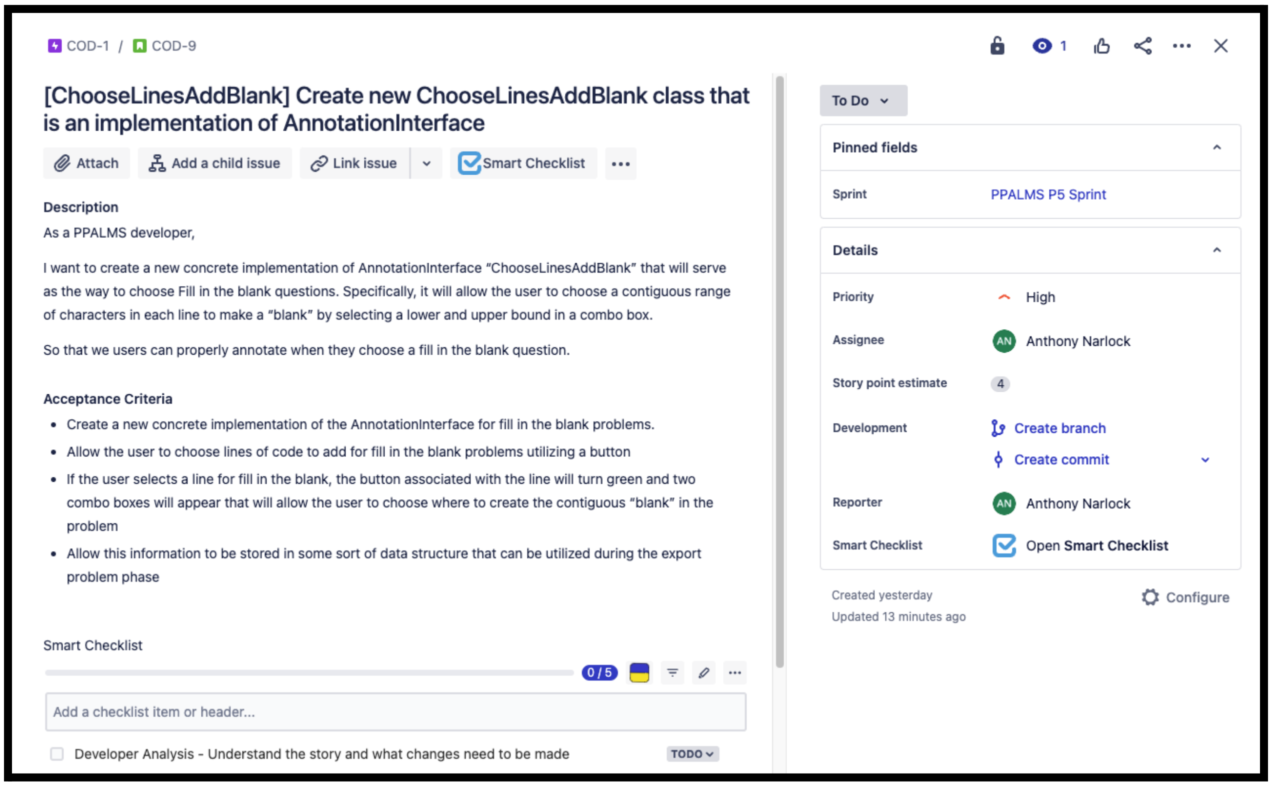
Sprint	PPALMS P5 Sprint

Details

Priority	Medium
Assignee	Anthony Narlock
Story point estimate	1
Development	Create branch Create commit
Reporter	Anthony Narlock
Smart Checklist	Open Smart Checklist

Created yesterday
Updated now

Configure


COD-1 / COD-9

[ChooseLinesAddBlank] Create new ChooseLinesAddBlank class that is an implementation of AnnotationInterface

Attach | Add a child issue | Link issue | Smart Checklist

Description

As a PPALMS developer,

I want to create a new concrete implementation of AnnotationInterface "ChooseLinesAddBlank" that will serve as the way to choose Fill in the blank questions. Specifically, it will allow the user to choose a contiguous range of characters in each line to make a "blank" by selecting a lower and upper bound in a combo box.

So that we users can properly annotate when they choose a fill in the blank question.

Acceptance Criteria

- Create a new concrete implementation of the AnnotationInterface for fill in the blank problems.
- Allow the user to choose lines of code to add for fill in the blank problems utilizing a button
- If the user selects a line for fill in the blank, the button associated with the line will turn green and two combo boxes will appear that will allow the user to choose where to create the contiguous "blank" in the problem
- Allow this information to be stored in some sort of data structure that can be utilized during the export problem phase

Smart Checklist 0 / 5

Add a checklist item or header...

- [ ] Developer Analysis - Understand the story and what changes need to be made TODO

To Do

Pinned fields

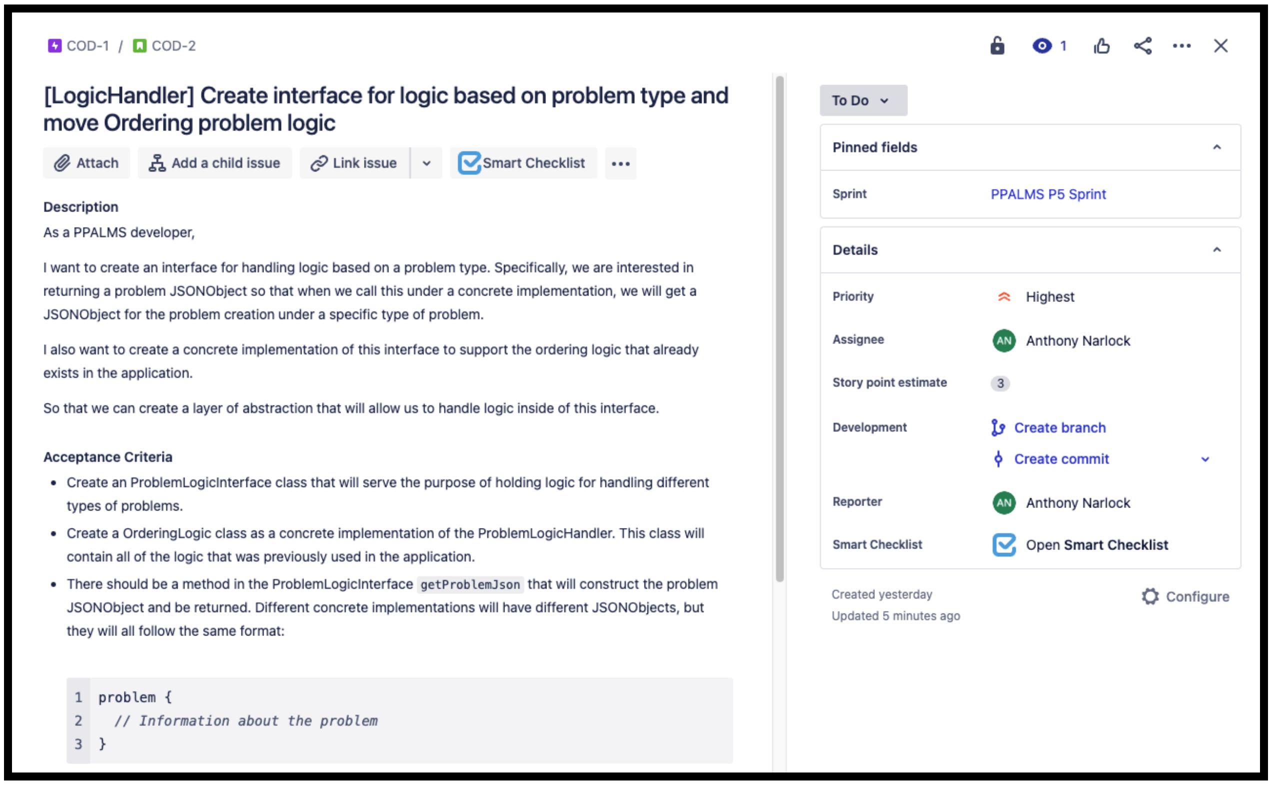
Sprint	PPALMS P5 Sprint

Details

Priority	High
Assignee	Anthony Narlock
Story point estimate	4
Development	Create branch Create commit
Reporter	Anthony Narlock
Smart Checklist	Open Smart Checklist

Created yesterday
Updated 13 minutes ago

Configure


COD-1 / COD-2

[LogicHandler] Create interface for logic based on problem type and move Ordering problem logic

Description

As a PPALMS developer,

I want to create an interface for handling logic based on a problem type. Specifically, we are interested in returning a problem JSONObject so that when we call this under a concrete implementation, we will get a JSONObject for the problem creation under a specific type of problem.

I also want to create a concrete implementation of this interface to support the ordering logic that already exists in the application.

So that we can create a layer of abstraction that will allow us to handle logic inside of this interface.

Acceptance Criteria

- Create an ProblemLogicInterface class that will serve the purpose of holding logic for handling different types of problems.
- Create a OrderingLogic class as a concrete implementation of the ProblemLogicHandler. This class will contain all of the logic that was previously used in the application.
- There should be a method in the ProblemLogicInterface `getProblemJson` that will construct the problem JSONObject and be returned. Different concrete implementations will have different JSONObjects, but they will all follow the same format:

```
problem {
  // Information about the problem
}
```

To Do

Pinned fields

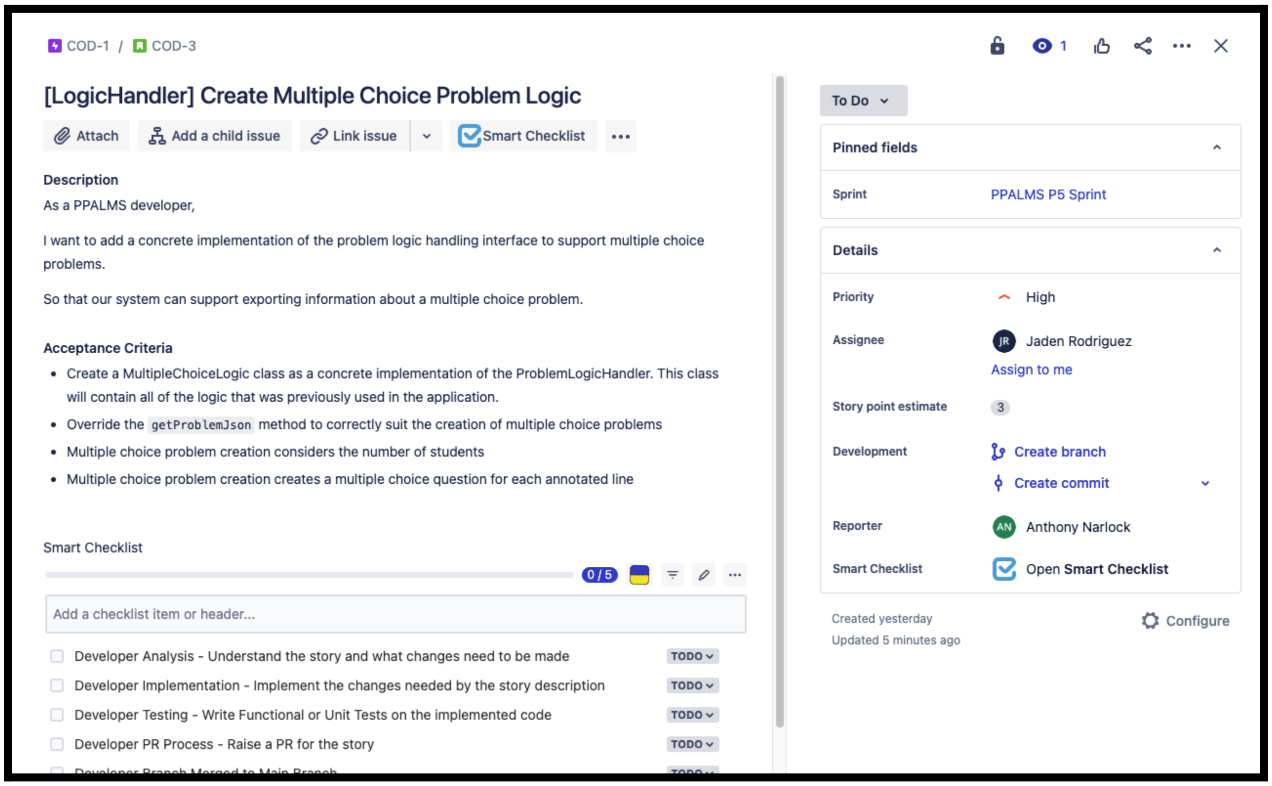
Sprint	PPALMS P5 Sprint

Details

Priority	Highest
Assignee	Anthony Narlock
Story point estimate	3
Development	Create branch Create commit
Reporter	Anthony Narlock
Smart Checklist	Open Smart Checklist

Created yesterday
Updated 5 minutes ago

Jaden was assigned COD-3 and COD-4. Screenshots of their descriptions, acceptance criteria, and other important details are shown below.


COD-1 / COD-3

[LogicHandler] Create Multiple Choice Problem Logic

Description

As a PPALMS developer,

I want to add a concrete implementation of the problem logic handling interface to support multiple choice problems.

So that our system can support exporting information about a multiple choice problem.

Acceptance Criteria

- Create a MultipleChoiceLogic class as a concrete implementation of the ProblemLogicHandler. This class will contain all of the logic that was previously used in the application.
- Override the `getProblemJson` method to correctly suit the creation of multiple choice problems
- Multiple choice problem creation considers the number of students
- Multiple choice problem creation creates a multiple choice question for each annotated line

Smart Checklist

0 / 5

Add a checklist item or header...

- ☐ Developer Analysis - Understand the story and what changes need to be made TODO
- ☐ Developer Implementation - Implement the changes needed by the story description TODO
- ☐ Developer Testing - Write Functional or Unit Tests on the implemented code TODO
- ☐ Developer PR Process - Raise a PR for the story TODO
- ☐ Developer Branch Merged to Main Branch TODO

To Do

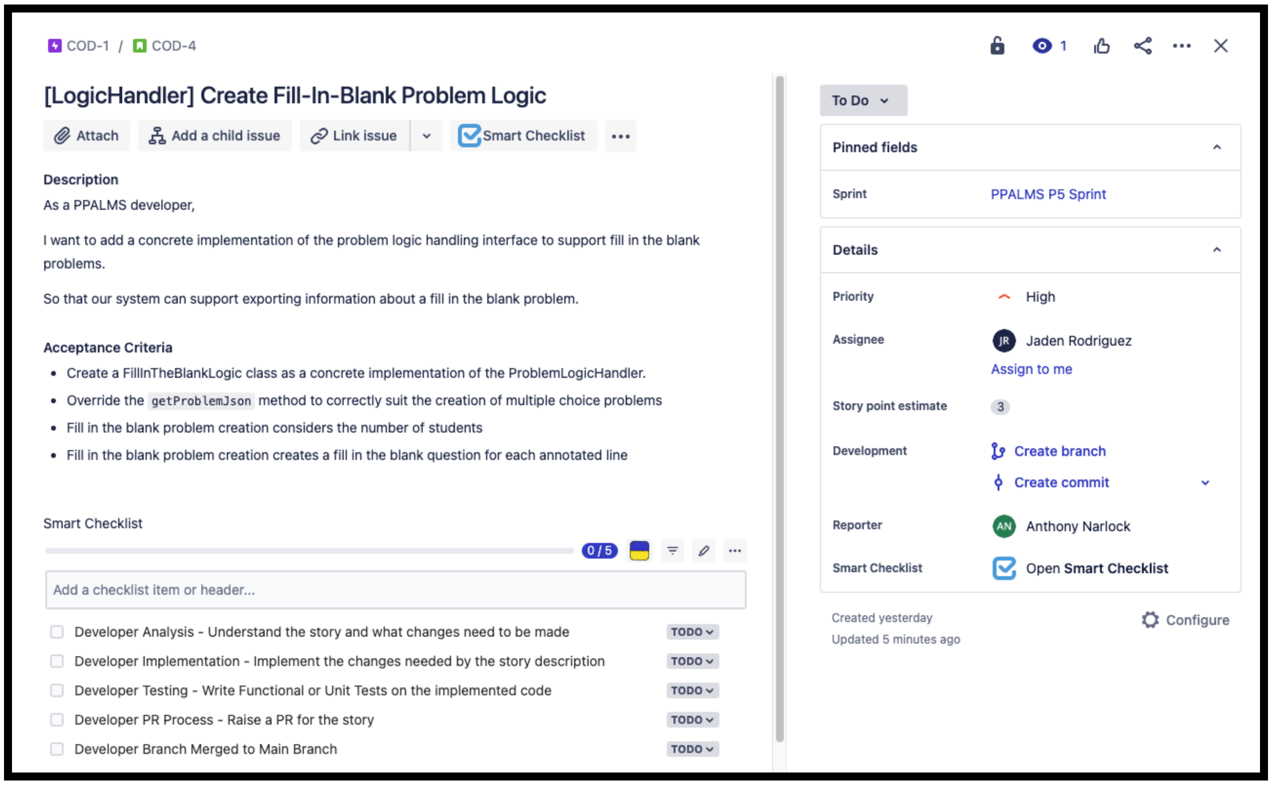
Pinned fields

Sprint	PPALMS P5 Sprint

Details

Priority	High
Assignee	Jaden Rodriguez Assign to me
Story point estimate	3
Development	Create branch Create commit
Reporter	Anthony Narlock
Smart Checklist	Open Smart Checklist

Created yesterday
Updated 5 minutes ago


COD-1 / COD-4

[LogicHandler] Create Fill-In-Blank Problem Logic

Attach | Add a child issue | Link issue | Smart Checklist

Description

As a PPALMS developer,

I want to add a concrete implementation of the problem logic handling interface to support fill in the blank problems.

So that our system can support exporting information about a fill in the blank problem.

Acceptance Criteria

- Create a FillInTheBlankLogic class as a concrete implementation of the ProblemLogicHandler.
- Override the `getProblemJson` method to correctly suit the creation of multiple choice problems
- Fill in the blank problem creation considers the number of students
- Fill in the blank problem creation creates a fill in the blank question for each annotated line

Smart Checklist 0 / 5

Add a checklist item or header...

- [ ] Developer Analysis - Understand the story and what changes need to be made TODO
- [ ] Developer Implementation - Implement the changes needed by the story description TODO
- [ ] Developer Testing - Write Functional or Unit Tests on the implemented code TODO
- [ ] Developer PR Process - Raise a PR for the story TODO
- [ ] Developer Branch Merged to Main Branch TODO

To Do

Pinned fields

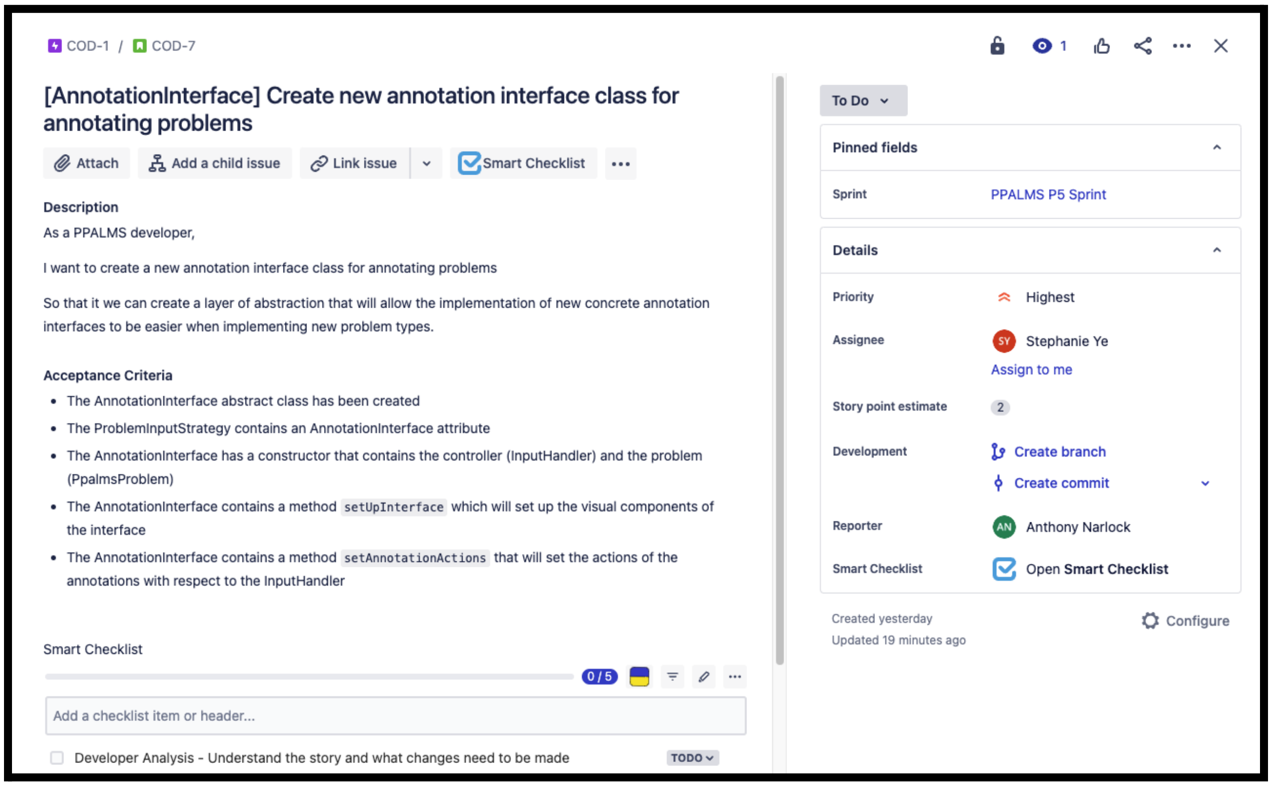
Sprint	PPALMS P5 Sprint

Details

Priority	High
Assignee	Jaden Rodriguez Assign to me
Story point estimate	3
Development	Create branch Create commit
Reporter	Anthony Narlock
Smart Checklist	Open Smart Checklist

Created yesterday
Updated 5 minutes ago
Configure

Stephanie was assigned COD-7 and COD-8. Screenshots of their descriptions, acceptance criteria, and other important details are shown below.


COD-1 / COD-7

[AnnotationInterface] Create new annotation interface class for annotating problems

Attach | Add a child issue | Link issue | Smart Checklist

Description

As a PPALMS developer,

I want to create a new annotation interface class for annotating problems

So that it we can create a layer of abstraction that will allow the implementation of new concrete annotation interfaces to be easier when implementing new problem types.

Acceptance Criteria

- The AnnotationInterface abstract class has been created
- The ProblemInputStrategy contains an AnnotationInterface attribute
- The AnnotationInterface has a constructor that contains the controller (InputHandler) and the problem (PpalmsProblem)
- The AnnotationInterface contains a method `setUpInterface` which will set up the visual components of the interface
- The AnnotationInterface contains a method `setAnnotationActions` that will set the actions of the annotations with respect to the InputHandler

Smart Checklist 0 / 5

Add a checklist item or header...

- [ ] Developer Analysis - Understand the story and what changes need to be made TODO

To Do

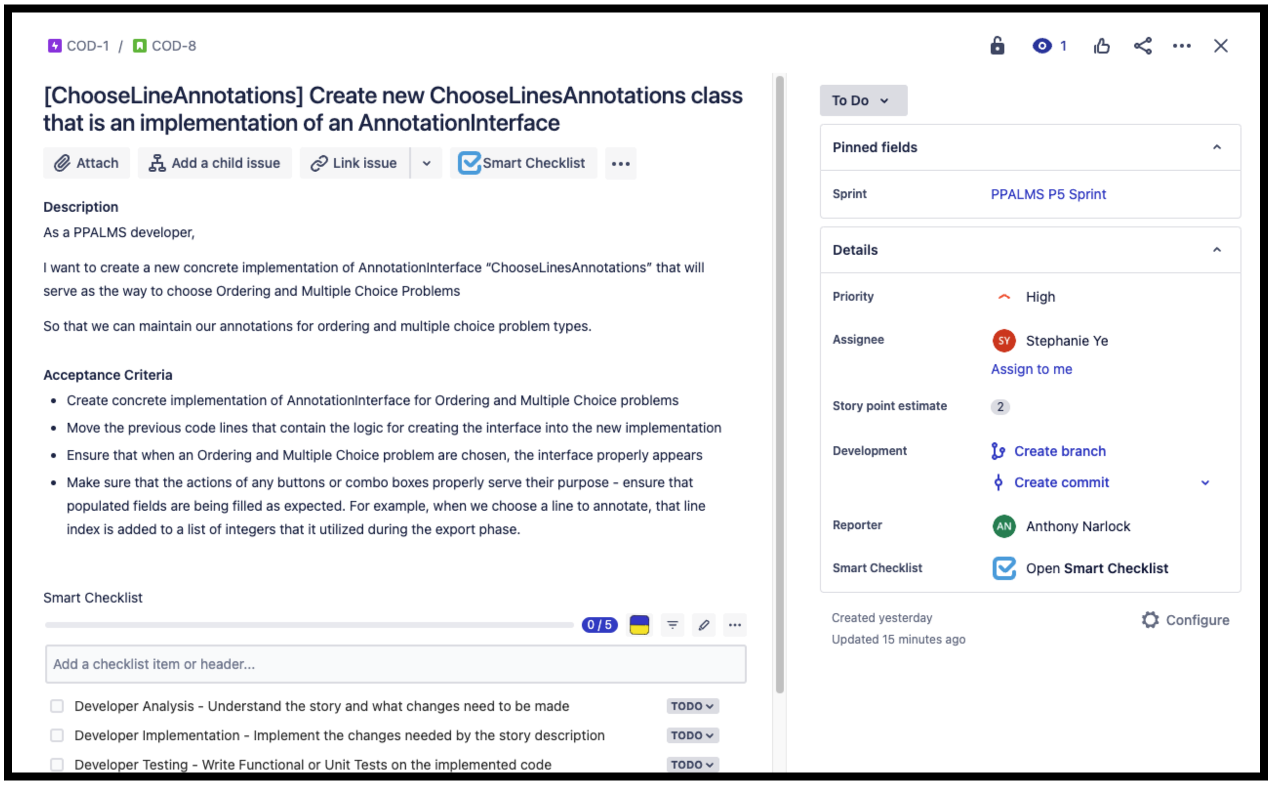
Pinned fields

Sprint	PPALMS P5 Sprint

Details

Priority	Highest
Assignee	Stephanie Ye Assign to me
Story point estimate	2
Development	Create branch Create commit
Reporter	Anthony Narlock
Smart Checklist	Open Smart Checklist

Created yesterday
Updated 19 minutes ago
Configure


COD-1 / COD-8

[ChooseLineAnnotations] Create new ChooseLinesAnnotations class that is an implementation of an AnnotationInterface

Attach | Add a child issue | Link issue | Smart Checklist

Description

As a PPALMS developer,

I want to create a new concrete implementation of AnnotationInterface "ChooseLinesAnnotations" that will serve as the way to choose Ordering and Multiple Choice Problems

So that we can maintain our annotations for ordering and multiple choice problem types.

Acceptance Criteria

- Create concrete implementation of AnnotationInterface for Ordering and Multiple Choice problems
- Move the previous code lines that contain the logic for creating the interface into the new implementation
- Ensure that when an Ordering and Multiple Choice problem are chosen, the interface properly appears
- Make sure that the actions of any buttons or combo boxes properly serve their purpose - ensure that populated fields are being filled as expected. For example, when we choose a line to annotate, that line index is added to a list of integers that it utilized during the export phase.

Smart Checklist 0 / 5

Add a checklist item or header...

- Developer Analysis - Understand the story and what changes need to be made TODO
- Developer Implementation - Implement the changes needed by the story description TODO
- Developer Testing - Write Functional or Unit Tests on the implemented code TODO

To Do

Pinned fields

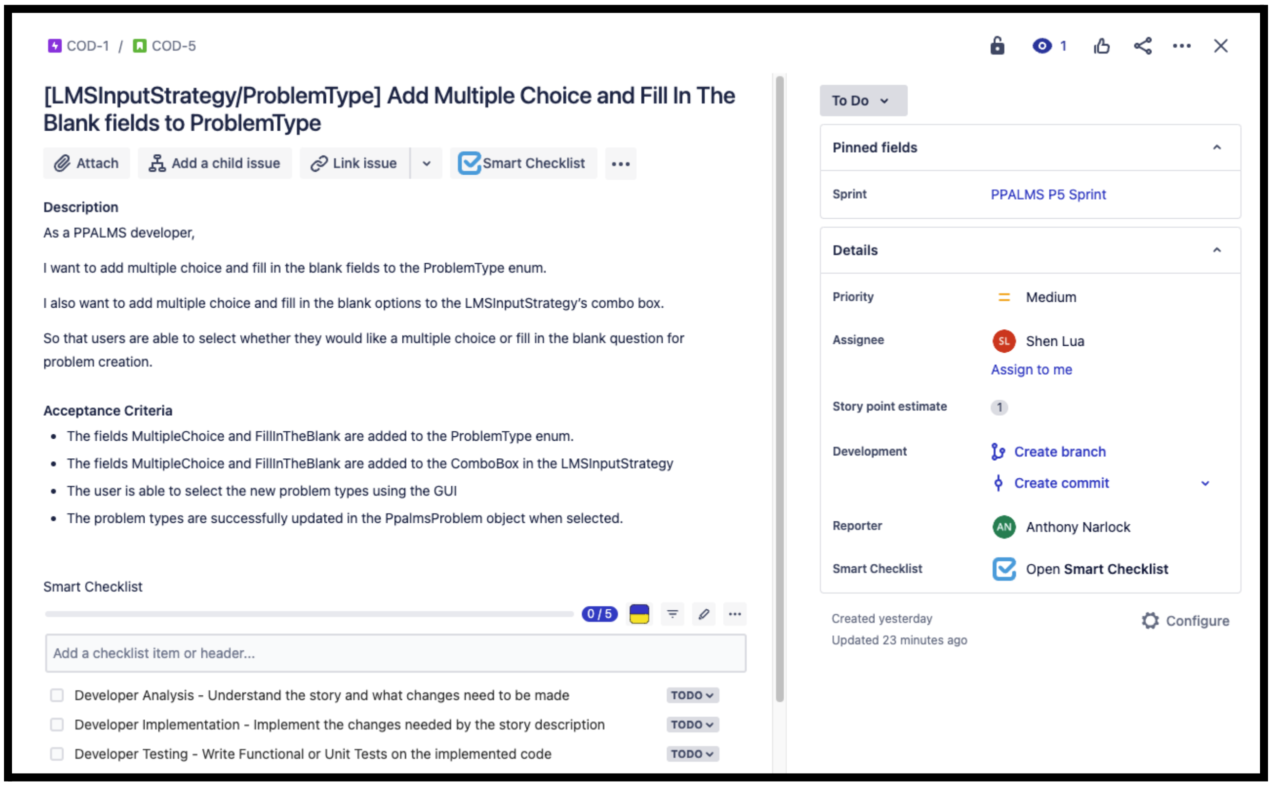
Sprint	PPALMS P5 Sprint

Details

Priority	High
Assignee	Stephanie Ye Assign to me
Story point estimate	2
Development	Create branch Create commit
Reporter	Anthony Narlock
Smart Checklist	Open Smart Checklist

Created yesterday
Updated 15 minutes ago
Configure

Shen was assigned COD-5 and COD-6. Screenshots of their descriptions, acceptance criteria, and other important details are shown below.


COD-1 / COD-5

[LMSInputStrategy/ProblemType] Add Multiple Choice and Fill In The Blank fields to ProblemType

Attach | Add a child issue | Link issue | Smart Checklist

Description

As a PPALMS developer,

I want to add multiple choice and fill in the blank fields to the ProblemType enum.

I also want to add multiple choice and fill in the blank options to the LMSInputStrategy's combo box.

So that users are able to select whether they would like a multiple choice or fill in the blank question for problem creation.

Acceptance Criteria

- The fields MultipleChoice and FillInTheBlank are added to the ProblemType enum.
- The fields MultipleChoice and FillInTheBlank are added to the ComboBox in the LMSInputStrategy
- The user is able to select the new problem types using the GUI
- The problem types are successfully updated in the PpalmsProblem object when selected.

Smart Checklist 0 / 5

Add a checklist item or header...

- Developer Analysis - Understand the story and what changes need to be made TODO
- Developer Implementation - Implement the changes needed by the story description TODO
- Developer Testing - Write Functional or Unit Tests on the implemented code TODO

To Do

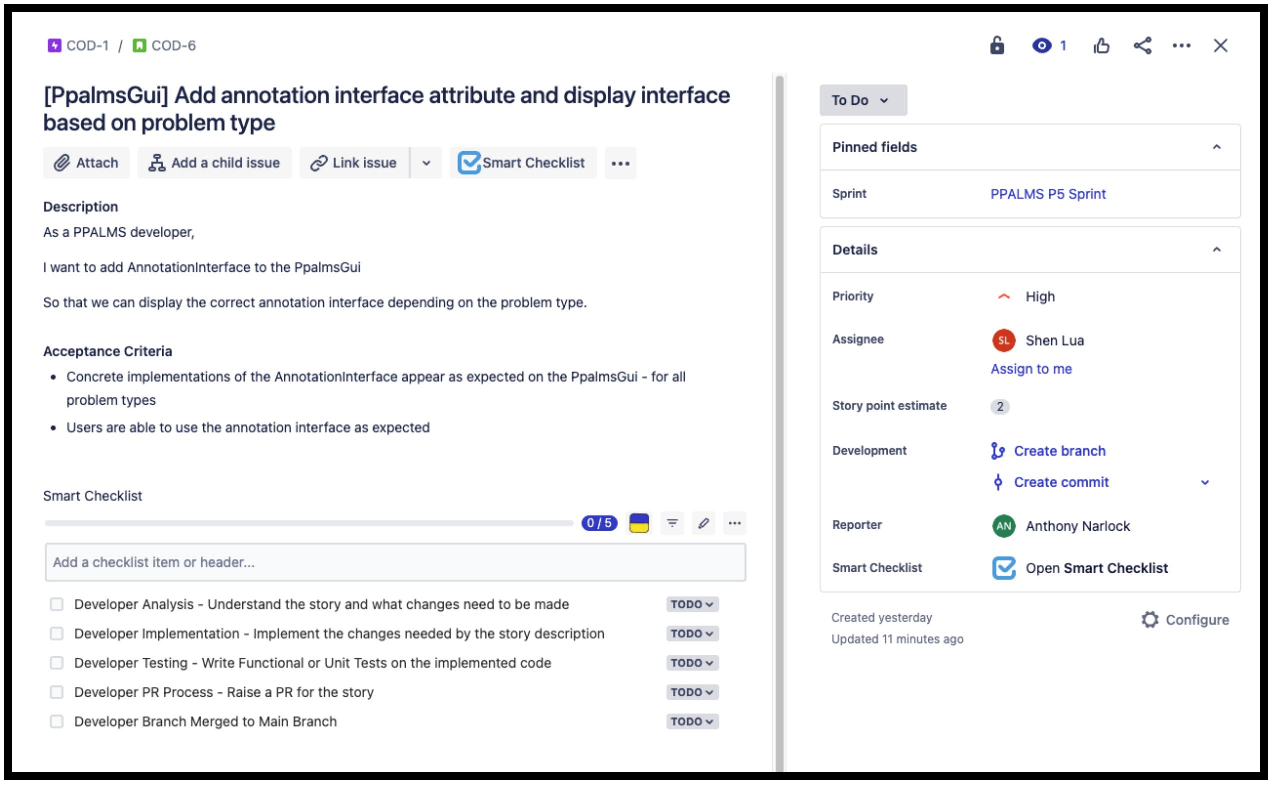
Pinned fields

Sprint	PPALMS P5 Sprint

Details

Priority	Medium
Assignee	Shen Lua Assign to me
Story point estimate	1
Development	Create branch Create commit
Reporter	Anthony Narlock
Smart Checklist	Open Smart Checklist

Created yesterday
Updated 23 minutes ago
Configure


COD-1 / COD-6

[PpalmsGui] Add annotation interface attribute and display interface based on problem type

Description

As a PPALMS developer,

I want to add AnnotationInterface to the PpalmsGui

So that we can display the correct annotation interface depending on the problem type.

Acceptance Criteria

- Concrete implementations of the AnnotationInterface appear as expected on the PpalmsGui - for all problem types
- Users are able to use the annotation interface as expected

Smart Checklist

0 / 5

Add a checklist item or header...

- [ ] Developer Analysis - Understand the story and what changes need to be made — TODO
- [ ] Developer Implementation - Implement the changes needed by the story description — TODO
- [ ] Developer Testing - Write Functional or Unit Tests on the implemented code — TODO
- [ ] Developer PR Process - Raise a PR for the story — TODO
- [ ] Developer Branch Merged to Main Branch — TODO

To Do

Pinned fields

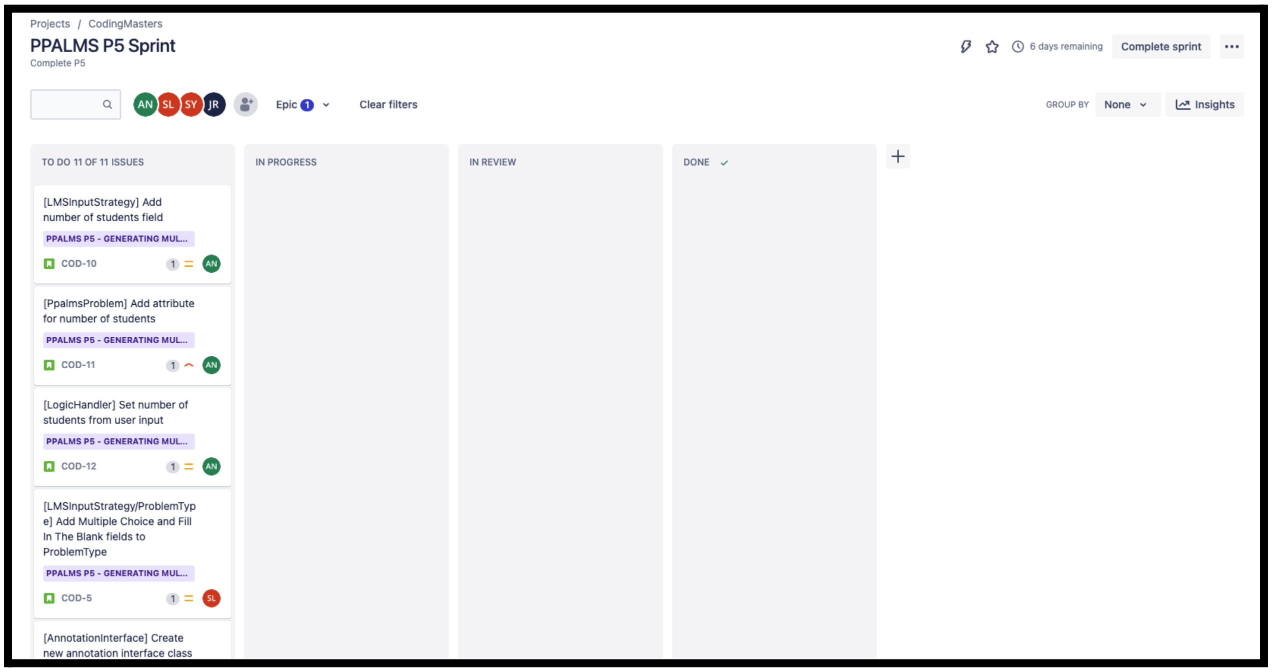
Sprint	PPALMS P5 Sprint

Details

Priority	High
Assignee	Shen Lua Assign to me
Story point estimate	2
Development	Create branch Create commit
Reporter	Anthony Narlock
Smart Checklist	Open Smart Checklist

Created yesterday
Updated 11 minutes ago

With these assignments, the sprint has begun. The SCRUM board is organized as follows. Note that we have an IN REVIEW column to show that the story needs to be reviewed by another developer before being completed as our definition of done states.


Projects / CodingMasters

PPALMS P5 Sprint

Complete P5

TO DO 11 OF 11 ISSUES

- [LMSInputStrategy] Add number of students field — PPALMS P5 - GENERATING MUL... — COD-10 — 1 — AN
- [PpalmsProblem] Add attribute for number of students — PPALMS P5 - GENERATING MUL... — COD-11 — 1 — AN
- [LogicHandler] Set number of students from user input — PPALMS P5 - GENERATING MUL... — COD-12 — 1 — AN
- [LMSInputStrategy/ProblemType] Add Multiple Choice and Fill In The Blank fields to ProblemType — PPALMS P5 - GENERATING MUL... — COD-5 — 1 — SL
- [AnnotationInterface] Create new annotation interface class

IN PROGRESS

IN REVIEW

DONE ✓

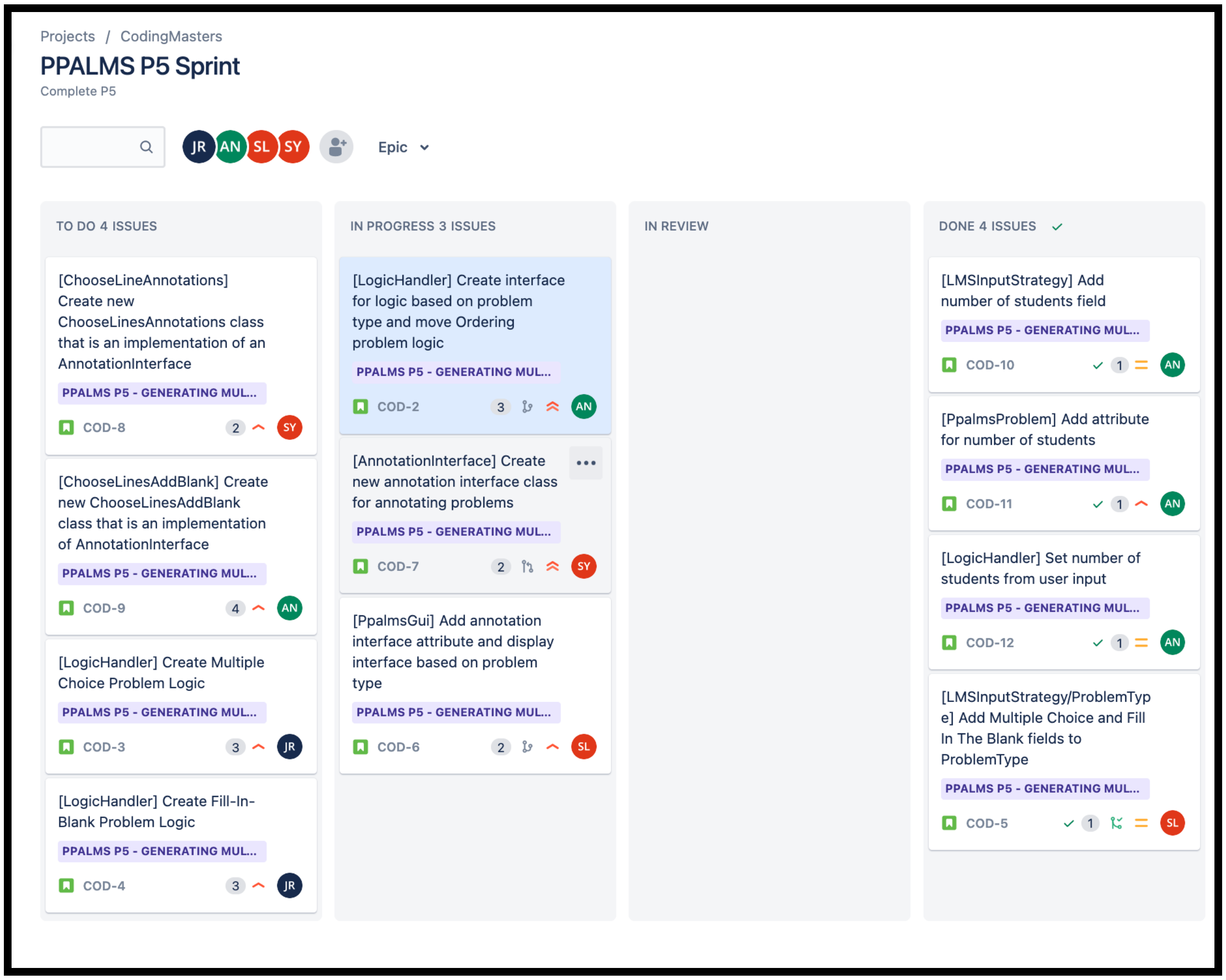
2. Daily Standup Meetings

2.1 First Daily SCRUM Meeting

Purpose	First Daily SCRUM - plan out day and identify obstacles
Date of Meeting, Time Duration	December 5th, 2022 5:15 PM - 5:30 PM (15 minutes)
Meeting Attendance	Anthony Narlock, Jaden Rodriguez, Shen Lua, Stephanie Ye

Stephanie is working on COD-7: Create new annotation interface class for annotating problems. Shen is working on COD-6: Add annotation interface attribute and display interface based on problem type. Anthony completed each of COD-5, COD-10, COD-11, and COD-12. Each of these tasks was reviewed by Jaden, and moved to the Done column. Anthony is now working on COD-2: Create interface for logic based on problem type and move Ordering problem logic. Jaden is working as SCRUM master; his tasks rely on COD-2. None of the in-progress tasks are assigned to him, but he is helping Anthony with COD-2 should he need it. We agreed that Stephanie and Shen are going to pair up to help one another with COD-6 and COD-7. We want the interfaces done at the same time so that we can see if they need any changes.

Assigning and helping one another with tasks is a method to mitigate obstacles and complete our daily work. We did not see a need to revise any of the in-progress stories. We decided as a team we did not need to plan a working session because everyone's progress was going accordingly.


Projects / CodingMasters

PPALMS P5 Sprint

Complete P5

JR AN SL SY Epic

TO DO 4 ISSUES

[ChooseLineAnnotations] Create new ChooseLinesAnnotations class that is an implementation of an AnnotationInterface
PPALMS P5 - GENERATING MUL...
COD-8 2 SY

[ChooseLinesAddBlank] Create new ChooseLinesAddBlank class that is an implementation of AnnotationInterface
PPALMS P5 - GENERATING MUL...
COD-9 4 AN

[LogicHandler] Create Multiple Choice Problem Logic
PPALMS P5 - GENERATING MUL...
COD-3 3 JR

[LogicHandler] Create Fill-In-Blank Problem Logic
PPALMS P5 - GENERATING MUL...
COD-4 3 JR

IN PROGRESS 3 ISSUES

[LogicHandler] Create interface for logic based on problem type and move Ordering problem logic
PPALMS P5 - GENERATING MUL...
COD-2 3 AN

[AnnotationInterface] Create new annotation interface class for annotating problems
PPALMS P5 - GENERATING MUL...
COD-7 2 SY

[PpalmsGui] Add annotation interface attribute and display interface based on problem type
PPALMS P5 - GENERATING MUL...
COD-6 2 SL

IN REVIEW

DONE 4 ISSUES ✓

[LMSInputStrategy] Add number of students field
PPALMS P5 - GENERATING MUL...
COD-10 ✓ 1 AN

[PpalmsProblem] Add attribute for number of students
PPALMS P5 - GENERATING MUL...
COD-11 ✓ 1 AN

[LogicHandler] Set number of students from user input
PPALMS P5 - GENERATING MUL...
COD-12 ✓ 1 AN

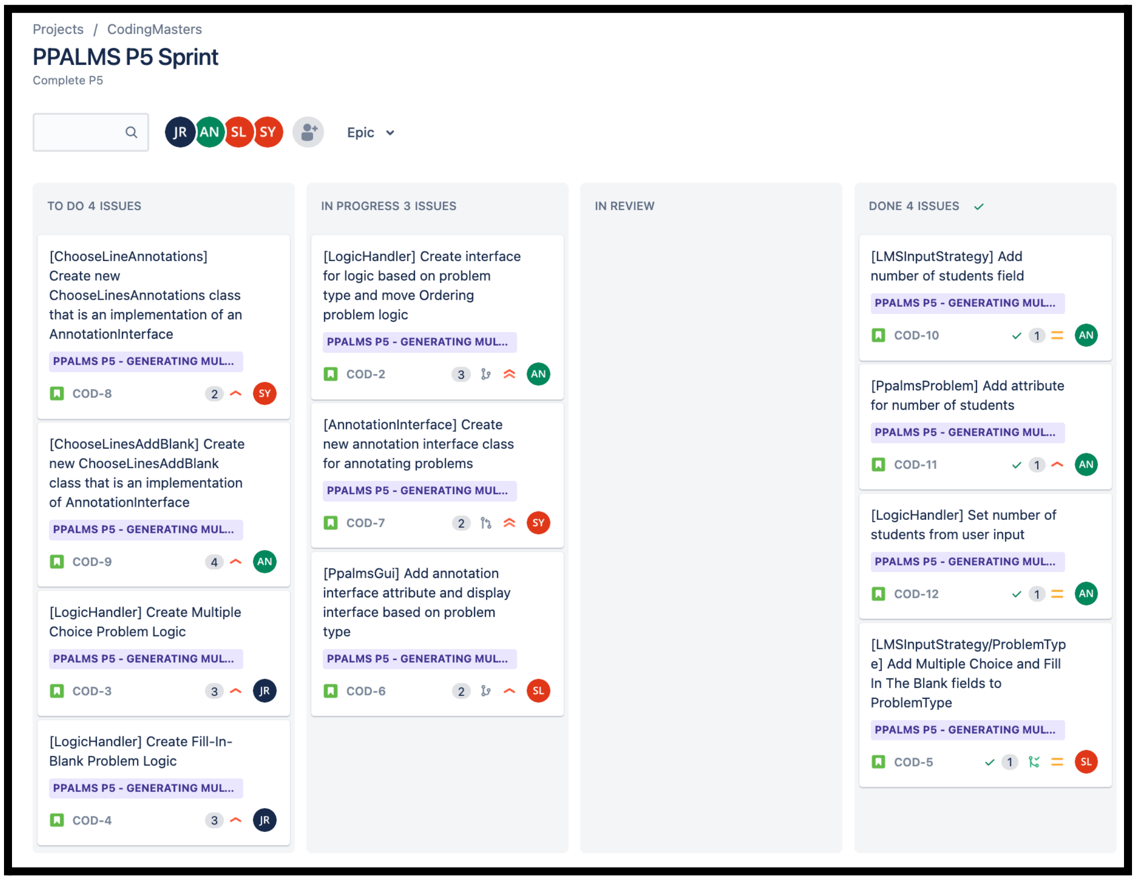
[LMSInputStrategy/ProblemType] Add Multiple Choice and Fill In The Blank fields to ProblemType
PPALMS P5 - GENERATING MUL...
COD-5 ✓ 1 SL

SCRUM board after meeting #1

2.2 Second Daily SCRUM Meeting

Purpose	Second Daily SCRUM - Check-in and Reassignment
Date of Meeting, Time Duration	December 6th, 2022 3:00 PM - 3:15 PM (15 minutes)
Meeting Attendance	Anthony Narlock, Jaden Rodriguez, Shen Lua, Stephanie Ye

Today Jaden asked each member about their progress on their tasks, and if they needed help. He updated the status of each task in the meeting notes. Stephanie hasn't made any progress on COD-7 as of today, and is getting help from Shen, who is working on a related task. Shen hasn't made any progress on COD-6 as of today because his task relies on Stephanie's task. He is helping Stephanie complete her task. Anthony has started COD-2. It is going well, and he expects it to be done in the next few days. Because his task relies on COD-2, Jaden has requested to help Anthony with his task so it can be done quicker.


Projects / CodingMasters

PPALMS P5 Sprint

Complete P5

JR AN SL SY Epic

TO DO 4 ISSUES

[ChooseLineAnnotations] Create new ChooseLinesAnnotations class that is an implementation of an AnnotationInterface
PPALMS P5 - GENERATING MUL...
COD-8 2 SY

[ChooseLinesAddBlank] Create new ChooseLinesAddBlank class that is an implementation of AnnotationInterface
PPALMS P5 - GENERATING MUL...
COD-9 4 AN

[LogicHandler] Create Multiple Choice Problem Logic
PPALMS P5 - GENERATING MUL...
COD-3 3 JR

[LogicHandler] Create Fill-In-Blank Problem Logic
PPALMS P5 - GENERATING MUL...
COD-4 3 JR

IN PROGRESS 3 ISSUES

[LogicHandler] Create interface for logic based on problem type and move Ordering problem logic
PPALMS P5 - GENERATING MUL...
COD-2 3 AN

[AnnotationInterface] Create new annotation interface class for annotating problems
PPALMS P5 - GENERATING MUL...
COD-7 2 SY

[PpalmsGui] Add annotation interface attribute and display interface based on problem type
PPALMS P5 - GENERATING MUL...
COD-6 2 SL

IN REVIEW

DONE 4 ISSUES ✓

[LMSInputStrategy] Add number of students field
PPALMS P5 - GENERATING MUL...
COD-10 ✓ 1 AN

[PpalmsProblem] Add attribute for number of students
PPALMS P5 - GENERATING MUL...
COD-11 ✓ 1 AN

[LogicHandler] Set number of students from user input
PPALMS P5 - GENERATING MUL...
COD-12 ✓ 1 AN

[LMSInputStrategy/ProblemType] Add Multiple Choice and Fill In The Blank fields to ProblemType
PPALMS P5 - GENERATING MUL...
COD-5 ✓ 1 SL

SCRUM board after meeting #2

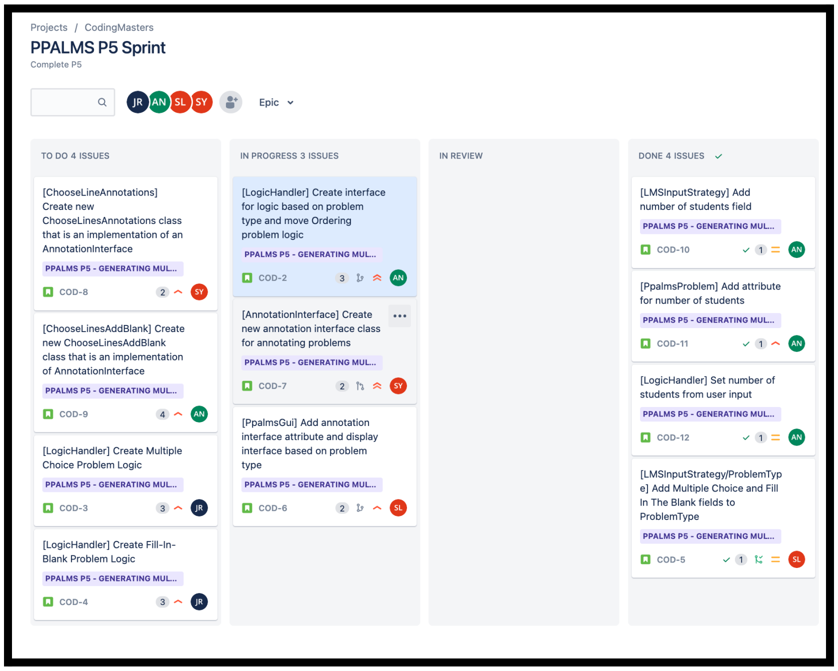
2.3 Third Daily SCRUM Meeting

Purpose	Third Daily SCRUM - Check-in and Reassignment
Date of Meeting, Time Duration	December 7th, 2022 5:15 PM - 5:30 PM (15 minutes)
Meeting Attendance	Anthony Narlock, Jaden Rodriguez, Shen Lua, Stephanie Ye

We will reserve this section for providing a description of what we spoke about during the daily standup meeting. We will talk about what each member of the team is working on and whether they need help or not.

Anthony implemented all of COD-2 and functionally tested it. Jaden is writing the JUnit tests for it. Anthony is now assigned COD-9: Create new ChooseLinesAddBlank class that is an implementation of AnnotationInterface. This task relies on Shen and Stephanie's tasks, so we are going to look at their progress. Stephanie said she didn't need help, and finished COD-7 and pushed it for review. Jaden moved a new task for her into the In-Progress column, COD-8: Create new ChooseLinesAnnotations class that is an implementation of an AnnotationInterface. Shen is working on COD-6 now that Stephanie has submitted her task for review. He will review her PR, and says he does not need help with COD-6.

We decided the stories do not need revision.


Projects / CodingMasters

PPALMS P5 Sprint

Complete P5

JR AN SL SY Epic

TO DO 4 ISSUES

- [ChooseLineAnnotations] Create new ChooseLinesAnnotations class that is an implementation of an AnnotationInterface — PPALMS P5 - GENERATING MUL... — COD-8 — 2 — SY
- [ChooseLinesAddBlank] Create new ChooseLinesAddBlank class that is an implementation of AnnotationInterface — PPALMS P5 - GENERATING MUL... — COD-9 — 4 — AN
- [LogicHandler] Create Multiple Choice Problem Logic — PPALMS P5 - GENERATING MUL... — COD-3 — 3 — JR
- [LogicHandler] Create Fill-In-Blank Problem Logic — PPALMS P5 - GENERATING MUL... — COD-4 — 3 — JR

IN PROGRESS 3 ISSUES

- [LogicHandler] Create interface for logic based on problem type and move Ordering problem logic — PPALMS P5 - GENERATING MUL... — COD-2 — 3 — AN
- [AnnotationInterface] Create new annotation interface class for annotating problems — PPALMS P5 - GENERATING MUL... — COD-7 — 2 — SY
- [PpalmsGui] Add annotation interface attribute and display interface based on problem type — PPALMS P5 - GENERATING MUL... — COD-6 — 2 — SL

IN REVIEW

DONE 4 ISSUES ✓

- [LMSInputStrategy] Add number of students field — PPALMS P5 - GENERATING MUL... — COD-10 — ✓ 1 — AN
- [PpalmsProblem] Add attribute for number of students — PPALMS P5 - GENERATING MUL... — COD-11 — ✓ 1 — AN
- [LogicHandler] Set number of students from user input — PPALMS P5 - GENERATING MUL... — COD-12 — ✓ 1 — AN
- [LMSInputStrategy/ProblemType] Add Multiple Choice and Fill In The Blank fields to ProblemType — PPALMS P5 - GENERATING MUL... — COD-5 — ✓ 1 — SL

SCRUM board after meeting #3

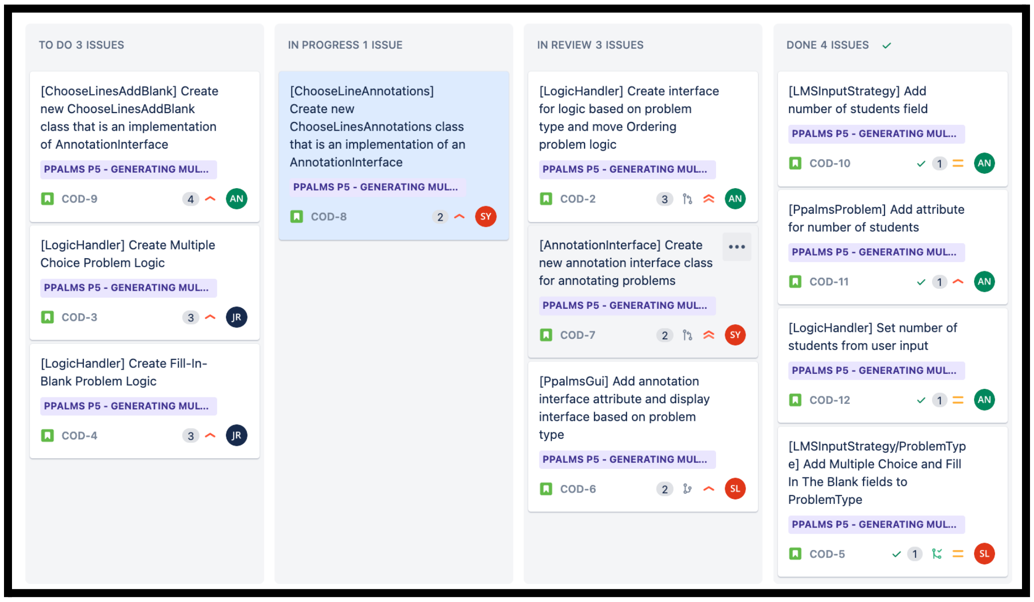
2.4 Fourth Daily SCRUM Meeting

Purpose	Fourth Daily SCRUM - Check-in and Reassignment
Date of Meeting, Time Duration	December 8th, 2022 3:00 PM - 3:15 PM (15 minutes)
Meeting Attendance	Anthony Narlock, Jaden Rodriguez, Shen Lua, Stephanie Ye

We will reserve this section for providing a description of what we spoke about during the daily standup meeting. We will talk about what each member of the team is working on and whether they need help or not. We will also take this time to mention anything that needs revising- stories need better descriptions, we need different ways to implement things, etc. We can also highlight that we plan to have working sessions on x days. We should also include a screenshot of what the scrum board looks like for more reference.

Stephanie is working on COD-8. She does not need help as of yet. Shen has finished COD-6, and is finalizing COD-7, and is almost done with it. He does not need help as of yet. Jaden is working on the unit tests for COD-2. He has requested some help from Anthony. Anthony's progress on COD-9 relies on the tasks of Shen and Stephanie, so he is currently helping Jaden finish the testing.

We've decided that we're going to allow the user to change the problem type without the corresponding input interface for now for review acceptance purposes until the input interface stories have been completed.



TO DO 3 ISSUES

- [ChooseLinesAddBlank] Create new ChooseLinesAddBlank class that is an implementation of AnnotationInterface — PPALMS P5 - GENERATING MUL... — COD-9 — 4 — AN
- [LogicHandler] Create Multiple Choice Problem Logic — PPALMS P5 - GENERATING MUL... — COD-3 — 3 — JR
- [LogicHandler] Create Fill-In-Blank Problem Logic — PPALMS P5 - GENERATING MUL... — COD-4 — 3 — JR

IN PROGRESS 1 ISSUE

- [ChooseLineAnnotations] Create new ChooseLinesAnnotations class that is an implementation of an AnnotationInterface — PPALMS P5 - GENERATING MUL... — COD-8 — 2 — SY

IN REVIEW 3 ISSUES

- [LogicHandler] Create interface for logic based on problem type and move Ordering problem logic — PPALMS P5 - GENERATING MUL... — COD-2 — 3 — AN
- [AnnotationInterface] Create new annotation interface class for annotating problems — PPALMS P5 - GENERATING MUL... — COD-7 — 2 — SY
- [PpalmsGui] Add annotation interface attribute and display interface based on problem type — PPALMS P5 - GENERATING MUL... — COD-6 — 2 — SL

DONE 4 ISSUES ✓

- [LMSInputStrategy] Add number of students field — PPALMS P5 - GENERATING MUL... — COD-10 — ✓ 1 — AN
- [PpalmsProblem] Add attribute for number of students — PPALMS P5 - GENERATING MUL... — COD-11 — ✓ 1 — AN
- [LogicHandler] Set number of students from user input — PPALMS P5 - GENERATING MUL... — COD-12 — ✓ 1 — AN
- [LMSInputStrategy/ProblemType] Add Multiple Choice and Fill In The Blank fields to ProblemType — PPALMS P5 - GENERATING MUL... — COD-5 — ✓ 1 — SL

SCRUM board after meeting #4

2.5 Fifth Daily SCRUM Meeting

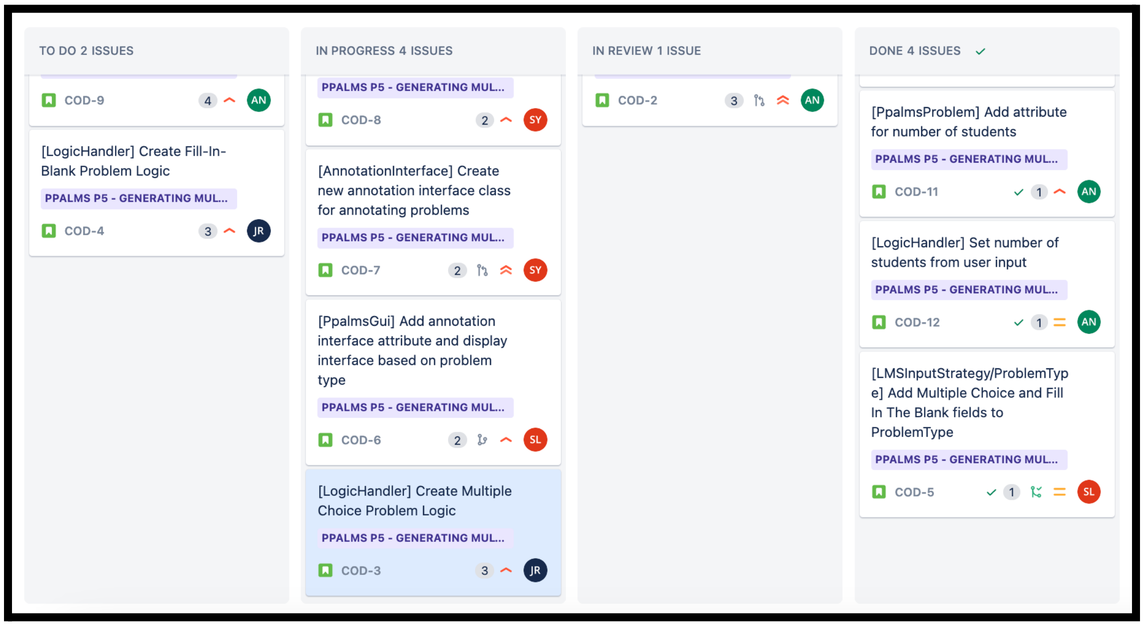
Purpose	Fifth Daily SCRUM - Check-in and Reassignment
Date of Meeting, Time Duration	December 9th, 2022 2:30 PM - 2:45 PM (15 minutes)
Meeting Attendance	Anthony Narlock, Jaden Rodriguez, Shen Lua, Stephanie Ye

Anthony is working on helping Jaden resolve the unit tests, and completed analysis of the COD-9 story. He is outlining the interface as he waits for the prerequisite stories to be resolved. Anthony and Jaden resolved an implementation ambiguity, and we decided that an initialized ordering problem which hasn't had its annotations set by the logic handler should permute all of the source lines, instead of those in its annotations list.

Jaden resolved the testing bugs in COD-2 and just needs to update the documentation and be reviewed to finish this task. He is assigned COD-3: [LogicHandler] Create Multiple Choice Problem Logic to work on next.

Stephanie and Shen are both working on their respective tasks from the last meeting, and all they have left is to write unit tests for their implementations. Anthony will then review and accept, and work on completing COD-9. Stephanie and Shen are helping one another write the unit tests.

No stories were refined, and one task was moved to the In Progress column.



TO DO 2 ISSUES

COD-9 4 AN

[LogicHandler] Create Fill-In-Blank Problem Logic
PPALMS P5 - GENERATING MUL...
COD-4 3 JR

IN PROGRESS 4 ISSUES

PPALMS P5 - GENERATING MUL...
COD-8 2 SY

[AnnotationInterface] Create new annotation interface class for annotating problems
PPALMS P5 - GENERATING MUL...
COD-7 2 SY

[PpalmsGui] Add annotation interface attribute and display interface based on problem type
PPALMS P5 - GENERATING MUL...
COD-6 2 SL

[LogicHandler] Create Multiple Choice Problem Logic
PPALMS P5 - GENERATING MUL...
COD-3 3 JR

IN REVIEW 1 ISSUE

COD-2 3 AN

DONE 4 ISSUES ✓

[PpalmsProblem] Add attribute for number of students
PPALMS P5 - GENERATING MUL...
COD-11 ✓ 1 AN

[LogicHandler] Set number of students from user input
PPALMS P5 - GENERATING MUL...
COD-12 ✓ 1 AN

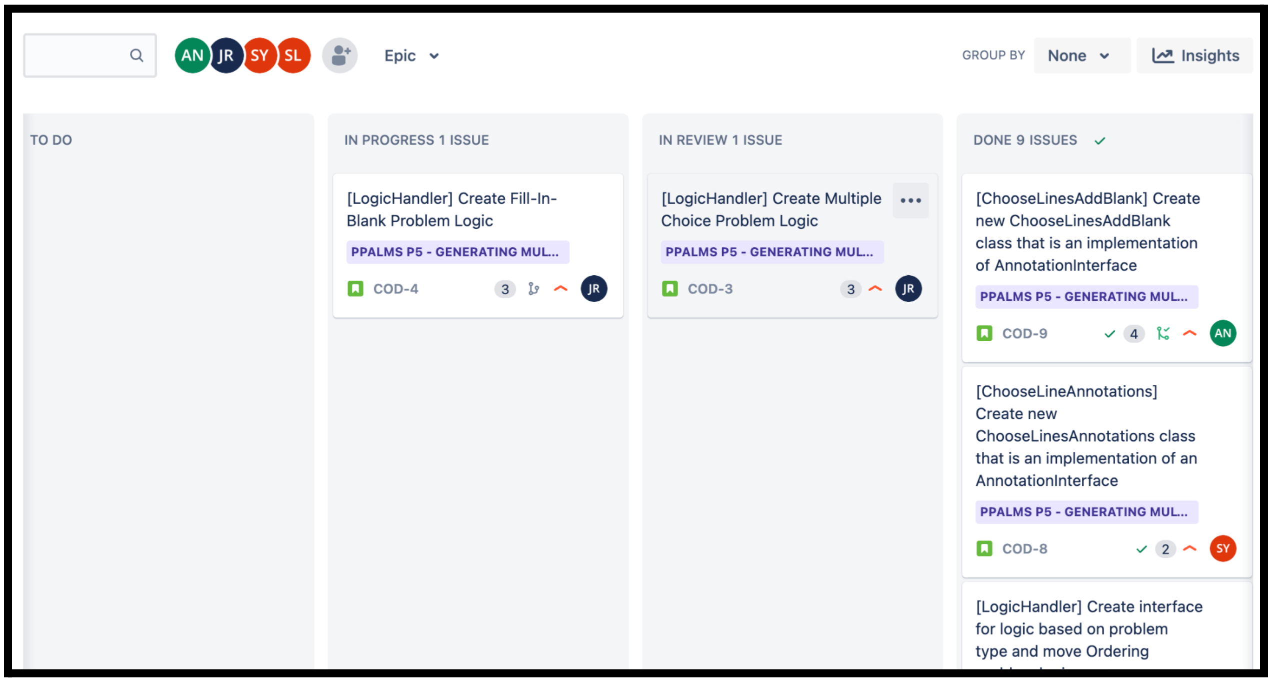
[LMSInputStrategy/ProblemType] Add Multiple Choice and Fill In The Blank fields to ProblemType
PPALMS P5 - GENERATING MUL...
COD-5 ✓ 1 SL

SCRUM board after meeting #5

2.6 Final Daily SCRUM Meeting

Purpose	Final Daily SCRUM - Check-in and Reassignment
Date of Meeting, Time Duration	December 12th, 2022 5:15 PM - 5:30 PM (15 minutes)
Meeting Attendance	Anthony Narlock, Jaden Rodriguez, Shen Lua, Stephanie Ye

Stephanie, Shen, and Anthony have completed their respective tasks and each were reviewed by another team member. Jaden has put the Multiple Choice implementation story up for review. Anthony is reviewing Jaden's pull request, and Stephanie and Shen are helping Jaden document his implementations. We decided as a group to segregate the number-of-students behavior from the rest of the problem creation interface, and add it in last. Our team has planned to complete the remainder of the tasks before our scheduled SCRUM review meeting tomorrow.



TO DO

IN PROGRESS 1 ISSUE

[LogicHandler] Create Fill-In-Blank Problem Logic
PPALMS P5 - GENERATING MUL...
COD-4 3 JR

IN REVIEW 1 ISSUE

[LogicHandler] Create Multiple Choice Problem Logic
PPALMS P5 - GENERATING MUL...
COD-3 3 JR

DONE 9 ISSUES ✓

[ChooseLinesAddBlank] Create new ChooseLinesAddBlank class that is an implementation of AnnotationInterface
PPALMS P5 - GENERATING MUL...
COD-9 ✓ 4 AN

[ChooseLineAnnotations] Create new ChooseLinesAnnotations class that is an implementation of an AnnotationInterface
PPALMS P5 - GENERATING MUL...
COD-8 ✓ 2 SY

[LogicHandler] Create interface for logic based on problem type and move Ordering

SCRUM board after the final daily standup meeting

3. Sprint Review Meeting

Purpose	Review work towards Sprint goal, mark completed stories, and plan next Sprint
Date of Meeting, Time Duration	December 13th, 2022 4:00 PM - 5:00 PM (60 minutes)
Meeting Attendance	Anthony Narlock, Jaden Rodriguez, Shen Lua, Stephanie Ye

In this sprint, we focused on the overarching goal of the epic, COD-1: PPALMS P5 - Generating Multiple Choice Problems, generating Fill-In-The-Blanks Problems, and adding a number-of-students attribute. We used it to determine stories to populate the product backlog with, and to move to the sprint backlog. We completed all 11 stories in our sprint backlog. Let's recall the definition of "done" that we settled on as a team. We define a story as "done" when we meet the following criteria: we have written passing unit tests for each branch or functionally tested each branch of the implementation of the story; the implementation has been reviewed and approved by another developer; the documentation, like JavaDoc comments in the source and unit tests, has been updated; and the execution reports for the tests have been written.

Each story met our definition of done, which was ensured by our use of smart checklists on each story, JIRA, and GitHub. For example, to merge a pull request, it first had to be reviewed by another team member. For each story the following things were completed: the story was tested, both functionally and with unit tests; the story was documented with JavaDocs and comments where necessary; the story had execution reports written on it; and the story was reviewed by at least one other team member. We completed each of the stories in the Sprint backlog, thus we met the Sprint goal and completed the Epic, COD-1.

For COD-2 [LogicHandler] Create Interface for Logic Based on ProblemType and move Ordering Problem logic, Anthony implemented the new ProblemCreationInterface so the PPALMS application can support creating multiple problems. Anthony added a JUnit test suite for the concrete OrderingCreation class that holds the implementation for the Ordering problem type. Jaden reviewed the pull request that was raised for this story before it was merged.

For COD-3: [LogicHandler] Create Multiple Choice Problem Logic, Jaden implemented a MultipleChoiceCreation class along with a corresponding JUnit test suite. Shen and Anthony helped make documentation like JavaDocs, and Anthony reviewed and resolved the pull request Jaden made to merge to the main branch.

For COD-4: [LogicHandler] Create Fill-In-Blank Problem Logic, Jaden implemented a FillInTheBlanksCreation class along with a corresponding JUnit test suite. Shen and Anthony helped make documentation like JavaDocs, and Anthony reviewed and resolved the pull request Jaden made to merge to the main branch.

For COD-5 : [LMSInputStrategy/ProblemType] Add Multiple Choice and Fill In the Blank fields to ProblemType, Shen added the fields MultipleChoice and FillInTheBlank to the ProblemType enum. Shen added the MultipleChoice and FillInTheblank problem type selection options to the ComboBox in the LMSInputStrategy. Shen created new unit tests and updated the already existing tests, and created JavaDocs for his test suite. Anthony reviewed and resolved the pull request Shen made to merge to the main branch.

For COD-6: [PpalmsGui] Add annotation interface attribute and display interface based on problem type. Shen added the implementation of this class to the PPALMS Gui class. Shen and Stephanie wrote the unit tests and documentations for it and finally, Anthony reviewed it and merged the COD-7 branch (which contained code pertaining to COD-6 and COD-8 since all these stories built on top of each other) into the main branch.

For COD-7: [AnnotationInterface] Create new annotation interface class for annotating problems, Stephanie implemented the AnnotationInterface class. Anthony helped review the story and resolved the pull request.

For COD-8: [ChooseLineAnnotations] Create new ChooseLinesAnnotations class that is an implementation of an AnnotationInterface, Stephanie and Shen implemented the ChooseLineAnnotations class.

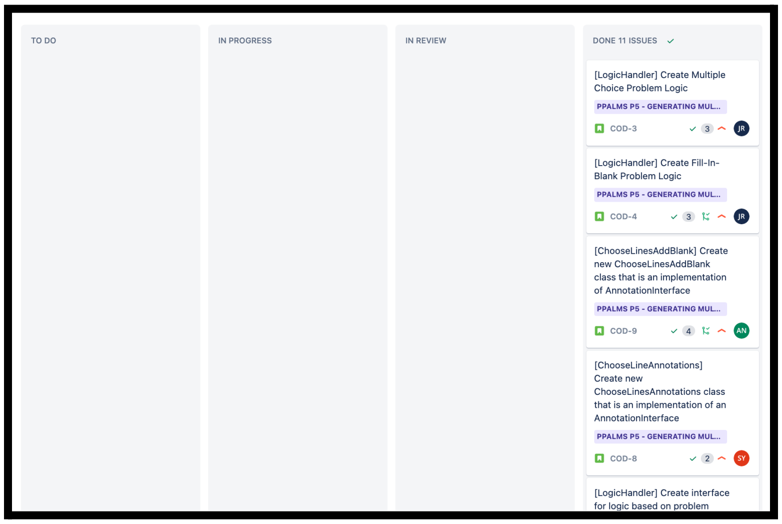
For COD-9 [ChooseLinesAddBlank] Create new ChooseLinesAddBlank class that is an implementation of AnnotationInterface. Anthony created a new concrete implementation of the AnnotationInterface that supports specifying annotations and indicating a lower and upper bound for an inserted blank in regards to a fill in the blank problem type. A test suite was created for this class before creating a pull request. Jaden reviewed and approved the pull request before it was merged.

For COD-10 [LMSInputStrategy] Add number of students field. Anthony created a number Of Students attribute selection to the LMSInputStrategy. This story allowed the user to indicate the number of students for their intended problem and contained proper input validation through the use of specialized Java Swing components. Additionally, Anthony created a test suite to complete this story. After raising a pull request, Shen and Stephanie both approved it and the code was merged.

For COD-11 [PpalmsProblem] Add attribute for number of students. Anthony added an attribute to the PpalmsProblem class that contains the number of students that will be served as input from the LMSInputStrategy. Anthony raised a pull request for this and Shen approved and merged it.

For COD-12 [LogicHandler] Set number of students from user input. Anthony added logic to the logic handler to handle the events of setting the number of students input from the view. Anthony added to the logic handler test suite to test this new functionality. Jaden approved the changes that Anthony made before it was merged.

The picture below shows the scrum board at the end of the sprint.



TO DO

IN PROGRESS

IN REVIEW

DONE 11 ISSUES ✓

[LogicHandler] Create Multiple Choice Problem Logic
PPALMS P5 - GENERATING MUL...
COD-3 ✓ 3 JR

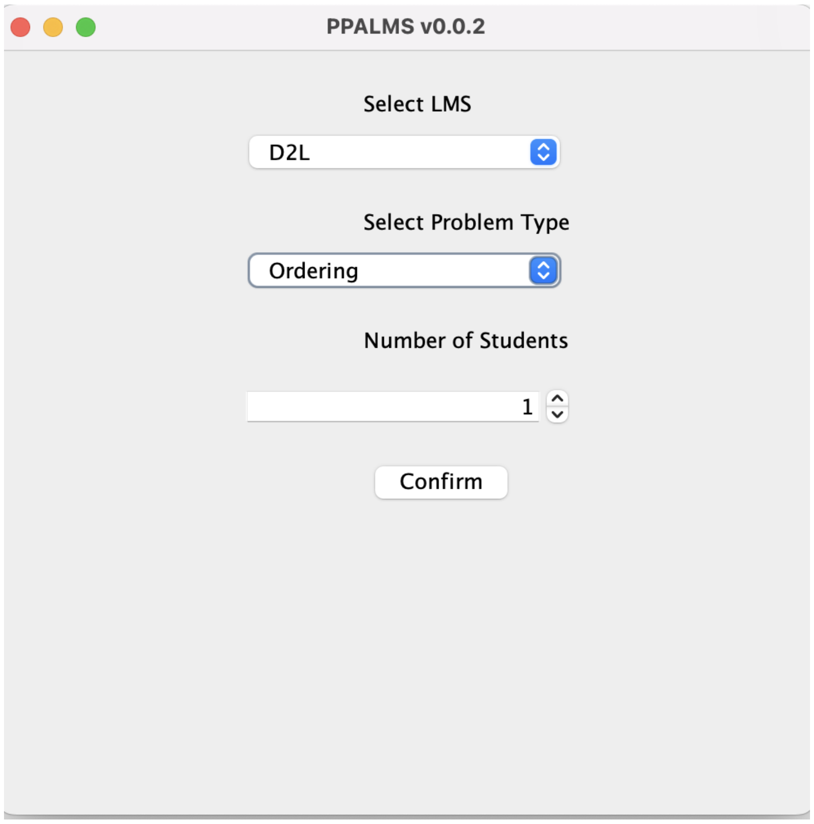
[LogicHandler] Create Fill-In-Blank Problem Logic
PPALMS P5 - GENERATING MUL...
COD-4 ✓ 3 JR

[ChooseLinesAddBlank] Create new ChooseLinesAddBlank class that is an implementation of AnnotationInterface
PPALMS P5 - GENERATING MUL...
COD-9 ✓ 4 AN

[ChooseLineAnnotations] Create new ChooseLinesAnnotations class that is an implementation of an AnnotationInterface
PPALMS P5 - GENERATING MUL...
COD-8 ✓ 2 SY

[LogicHandler] Create interface for logic based on problem

The following screenshots show new implementations from this story in our user interface. These are things that we demonstrated as a group during this meeting.



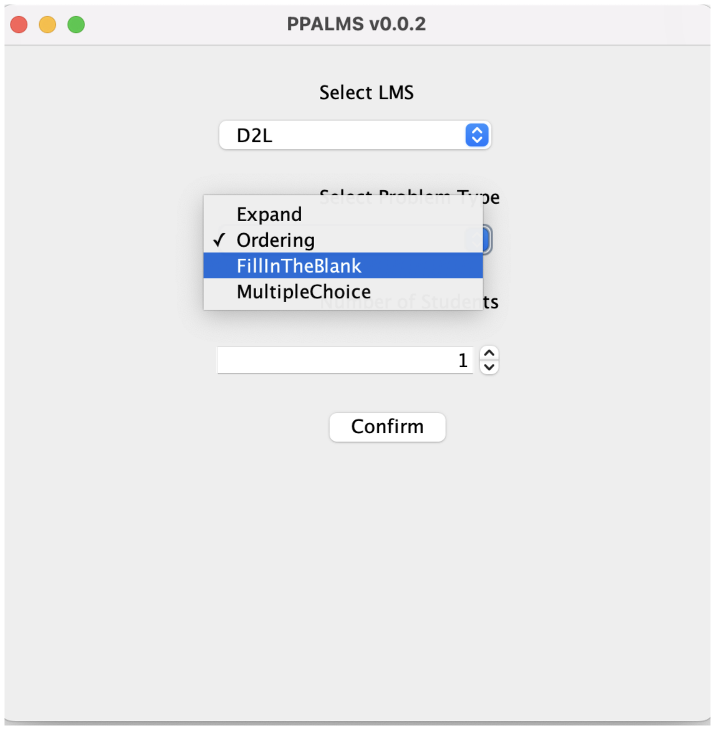
PPALMS v0.0.2

Select LMS: D2L

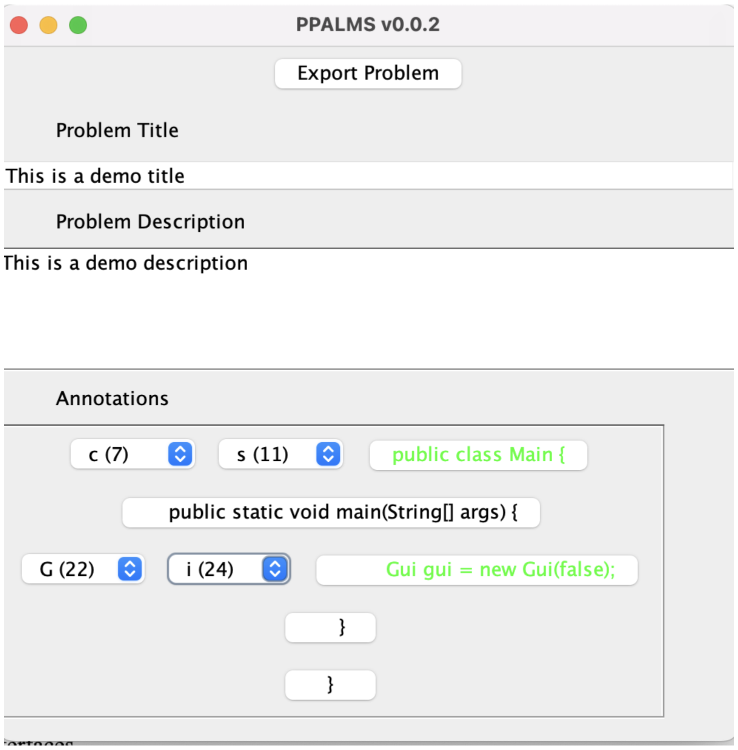
Select Problem Type: Ordering

Number of Students: 1

The number of students field for the LMSInputStrategy



The PPALMS now supports fill in the blank and multiple choice problems for creation

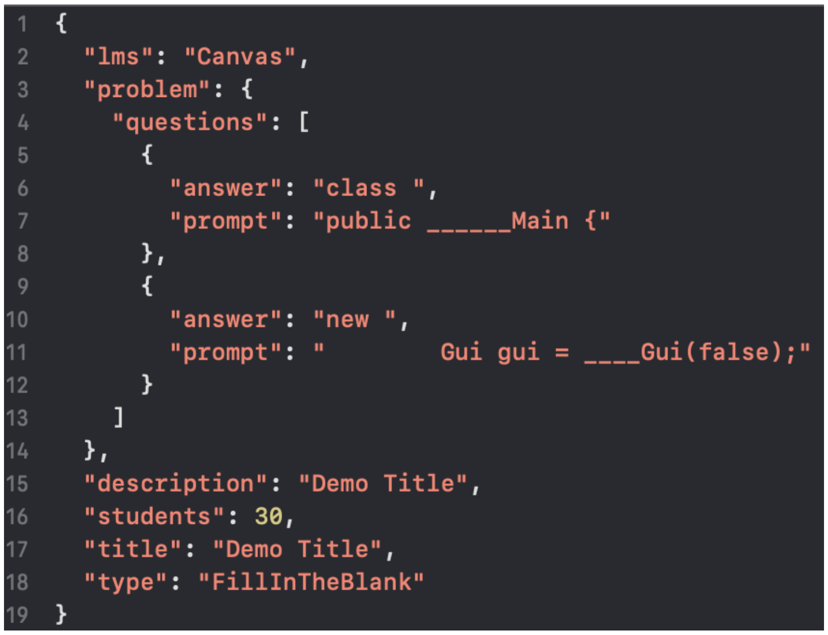


This is a demo title

This is a demo description

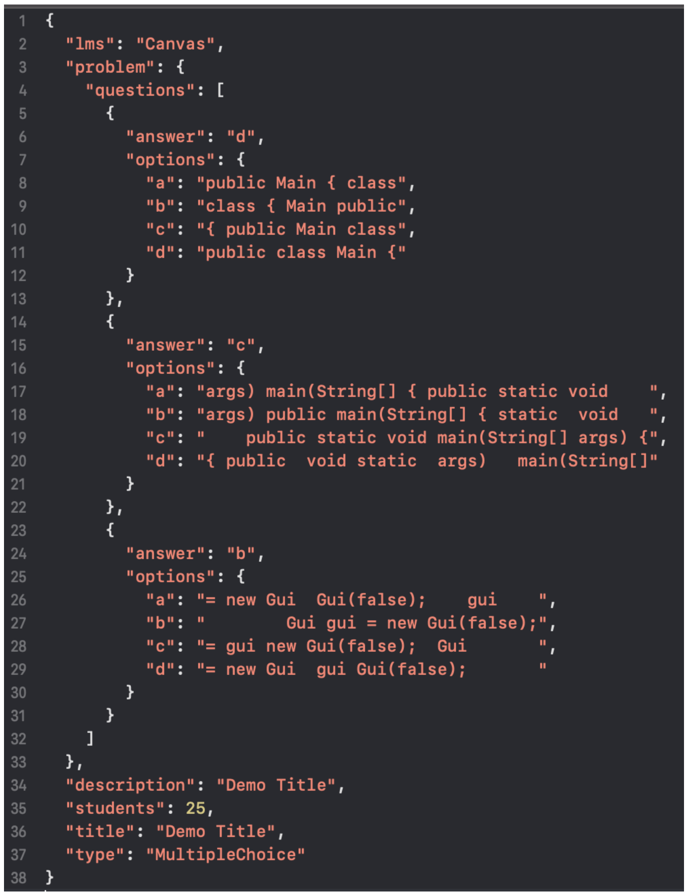
```
public class Main {
public static void main(String[] args) {
Gui gui = new Gui(false);
}
}
```

The annotation interface for fill in the blank questions



```json
{
  "lms": "Canvas",
  "problem": {
    "questions": [
      {
        "answer": "class ",
        "prompt": "public ______Main {"
      },
      {
        "answer": "new ",
        "prompt": "    Gui gui = ____Gui(false);"
      }
    ]
  },
  "description": "Demo Title",
  "students": 30,
  "title": "Demo Title",
  "type": "FillInTheBlank"
}
```

Example output JSON file for FillInTheBlank ProblemType



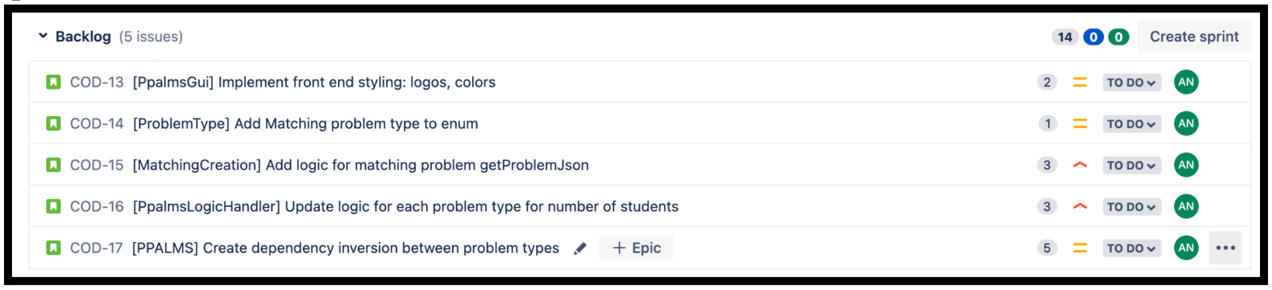
```json
{
  "lms": "Canvas",
  "problem": {
    "questions": [
      {
        "answer": "d",
        "options": {
          "a": "public Main { class",
          "b": "class { Main public",
          "c": "{ public Main class",
          "d": "public class Main {"
        }
      },
      {
        "answer": "c",
        "options": {
          "a": "args) main(String[] { public static void    ",
          "b": "args) public main(String[] { static  void   ",
          "c": "    public static void main(String[] args) {",
          "d": "{ public  void static  args)   main(String[]"
        }
      },
      {
        "answer": "b",
        "options": {
          "a": "= new Gui  Gui(false);    gui    ",
          "b": "        Gui gui = new Gui(false);",
          "c": "= gui new Gui(false);  Gui       ",
          "d": "= new Gui  gui Gui(false);       "
        }
      }
    ]
  },
  "description": "Demo Title",
  "students": 25,
  "title": "Demo Title",
  "type": "MultipleChoice"
}
```

Example output JSON file for MultipleChoice ProblemType

We discussed a number of items to add to the product backlog. In a future sprint, we would add these to the sprint backlog as our epic.

One item to add to the product backlog is to make the user interface more visually appealing to the user. This could be done by adding logos, colors, and other styling, as well as using more dynamic input interfaces. Another item we discussed adding is adding a “matching” type of Parson’s problem. Furthermore, we discussed an item to update the logic so that the number of students attribute is used to generate a problem assignment. Another item we discussed is refactoring our code base so that it follows more of the SOLID principles. This would make it easier to implement and accommodate new problem types.

The following picture shows the sprint backlog for tasks that will be completed in our next sprint.



COD-13	[PpalmsGui] Implement front end styling: logos, colors	2	TO DO	AN
COD-14	[ProblemType] Add Matching problem type to enum	1	TO DO	AN
COD-15	[MatchingCreation] Add logic for matching problem getProblemJson	3	TO DO	AN
COD-16	[PpalmsLogicHandler] Update logic for each problem type for number of students	3	TO DO	AN
COD-17	[PPALMS] Create dependency inversion between problem types	5	TO DO	AN

4. Sprint Retrospective

Purpose	Honest assessment of the work that was done during the sprint.
Date of Meeting, Time Duration	December 13th, 2022 5:00 PM - 6:00 PM (60 minutes)
Meeting Attendance	Anthony Narlock, Jaden Rodriguez, Shen Lua, Stephanie Ye

4.1 What Went Well

We did a good job at sticking to our scrum meeting, scrum retrospective meeting, and scrum review meeting times. Our definition of done helped us to satisfy our requirements and not be forgetful or make mistakes. The implemented features functioned as expected and they have corresponding unit tests and documentation. One of Jaden's unit tests caught an error made in the FillInTheBlankCreation class where he accidentally used a hyphen instead of an underscore. This indicates that the testing was a good criterion to ensure correctness and coverage, since it exposed human errors that could have otherwise been overlooked.

It is also worth noting that our end product is very complete and functional. We didn't have noticeable buggy or missing behavior from our requirements, and the output was very intuitive and readable.

4.2 What Didn't Go Well

We had a lot of scheduling conflicts that resulted in "hurry up" and "waiting" situations. We planned out PPALMS working sessions but we sometimes did not follow through with those or used that time to work on other projects. This resulted in us working on the PPALMS system when not everyone was able to be present which caused a great deal of stress and frustration.

We also assigned different stories that were dependent on each other to different people, which didn't go well as the sprint played out. When one person was stuck or confused, the other person was blocked from working and had to wait for a while. Our definition of done further exacerbated this problem, because completion of testing, documentation, and review added to the wait time of those working on dependent stories.

4.3 What Could Be Improved

When we held planning or retrospective meetings, we tended to get sidetracked a lot. So what could have been an hour meeting ended up being upwards of two hours because we could not focus on the task at hand. One way of resolving this might be to change our meeting setting, our to have a pre-planned skeleton of what our meeting should look like to use as a guide and remain timely.

Another thing to improve would be the story conflicts we discussed in 4.2. We could mitigate this by evaluating the dependencies of stories with one another. We could even assign a score or group to each story similarly to how we did for the time required for each story. This would help us to assign stories in a way that would let our group be more efficient and concurrent.

4.4 Kudos!

From Stephanie: Kudos to Anthony for being a workaholic. He kept the project going and expected updates from everyone so we were clear on what progress we've all made.

From Shen : Kudos to Jaden for implementing the logic behind the problem permutations and mutation generation for the problem types ordering, fill-in-the-blanks and multiple choice.

From Jaden: Kudos to Anthony for helping us make a professional and complete product. Having frequent meetings and discussions helped me to get this project done much better than the last, and to resolve confusions or issues with my tasks. Thanks to Shen and Stephanie for helping me with my documentation and execution reports.

From Anthony: Kudos to Jaden for creating very creative logic for the multiple choice creation and fill-in-the-blank creation!